

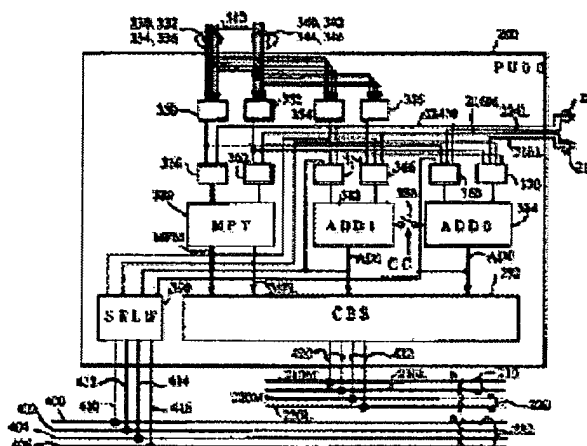
PARALLEL PROCESSING PROCESSOR, ITS PROCESSING UNIT AND OPERATING METHOD FOR THE PARALLEL PROCESSING PROCESSOR

Publication number: JP7064789
Publication date: 1995-03-10
Inventor: NAKASE YASUNOBU
Applicant: MITSUBISHI ELECTRIC CORP
Classification:
- international: G06F9/38; G06F9/38; (IPC1-7): G06F9/38
- European:
Application number: JP19930210783 19930825
Priority number(s): JP19930210783 19930825

Report a data error here

Abstract of JP7064789

PURPOSE: To provide a general purpose parallel processing processor which can easily be controlled. **CONSTITUTION:** Each of the plural processing units of the processor includes computing elements 380, 382 and 384, selectors 360 to 370 for giving the high/low order bits of input data buses 216 and 224 from another unit to these computing elements and a cross bar switch 392 for outputting the outputs of the computing elements 380, 382 and 384 to optional one of data buses 210 and 220 to another unit. The connection of respective selectors is previously set so as to execute various arithmetic operation. On the other hand, data switching between with another processing unit by way of the data bus realizes various arithmetic operation as the whole processor. The constitution of each processing unit is the same, easily laid out and can be controlled by interchangeable control instruction.



Data supplied from the esp@cenet database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号

特開平7-64789

(43) 公開日 平成 7 年 (1995) 3 月 10 日

(51) Int.Cl.⁶

G 0 6 F 9/38

識別記号

3 7 0 A

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数19 O L (全 29 頁)

(21) 出願番号 特願平5-210783

(22) 出願日 平成 5 年 (1993) 8 月 25 日

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目 2 番 3 号

(72) 発明者 中瀬 泰伸

兵庫県伊丹市瑞原 4 丁目 1 番地 三菱電機

株式会社エル・エス・アイ研究所内

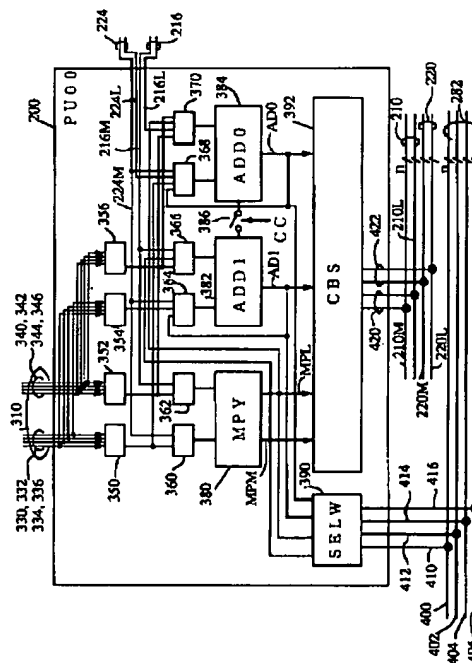
(74) 代理人 弁理士 深見 久郎 (外 3 名)

(54) 【発明の名称】 並列処理プロセッサおよびそのプロセッシングユニットならびにこの並列処理プロセッサの動作方法

(57) 【要約】

【目的】 簡単に制御可能な、汎用的並列処理プロセッサを提供する。

【構成】 プロセッサの複数のプロセッシングユニットの各々は、演算器 380、382、384 と、他のユニットからの入力データバス 216、224 の上位 n ビットまたは下位 n ビットをこれら演算器に与えるためのセクタ 360 ~ 370 と、演算器 380、382、384 の出力を他のユニットへのデータバス 210、220 の任意のものに出力するためのクロスバスイッチ 392 とを含む。各セクタの接続を予め設定することにより様々な演算を実行できる。また他のプロセッシングユニットとのデータバスを介したデータ交換によりプロセッサ全体として多様な演算を実現できる。各プロセッシングユニットの構造は同一でレイアウト容易であり、互換性のある制御命令で制御できる。



1

【特許請求の範囲】

【請求項1】 それぞれ複数個の入力を有し、与えられるデータの間に所定の演算を行なって結果を出力するための複数個の演算手段と、

複数の単方向入力データバスに接続され、前記複数個の演算手段の各入力ごとに、前記複数の単方向入力データバスのうちのいずれか1つを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータの一部を前記入力に与えるための入力データバス選択手段と、

前記複数個の演算手段の出力に接続された入力と、前記単方向入力データバスと同じ数の単方向出力データバスに接続された出力とを有し、前記演算手段の出力の各々を、前記単方向出力データバスのいずれかに出力するための出力データバス選択手段と、

前記複数個の演算手段により所望の複合演算を実現するために、前記入力データバス選択手段と、前記出力データバス選択手段とによるデータの経路を制御するための制御手段とを含む、並列処理のためのプロセッシングユニット。

【請求項2】 前記複数個の演算手段が、

2つの n ビット幅の入力を有し、与えられる2つのデータを乗算して $2n$ ビットの結果を出力する乗算器と、各々が2つの n ビット幅の入力を有し、与えられる2つのデータを加算して n ビット幅の結果を出力する2つの加算器とを含む、請求項1に記載の並列処理のためのプロセッシングユニット。

【請求項3】 前記複数の単方向入力データバスおよび前記複数の単方向出力データバスの各々は $2n$ ビット幅を有し、

前記入力データバス選択手段は、前記複数個の演算手段の各入力ごとに前記複数の単方向入力データバスのうちのいずれか1つを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータの上位または下位の n ビットを前記入力に与えるための手段を含み、

前記出力データバス選択手段は、前記演算手段の出力の各々を、前記単方向出力データバスの任意のいずれかの上位 n ビットまたは下位 n ビットまたはその双方に出力可能とするための手段を含む、請求項2に記載の並列処理のためのプロセッシングユニット。

【請求項4】 前記2つの加算器の一方はキャリー出力を有し、

前記2つの加算器の他方はキャリー入力を有し、

さらに、前記キャリー出力と前記キャリー入力を可制御的に断続するための手段を含む、請求項2に記載の並列処理のためのプロセッシングユニット。

【請求項5】 前記入力データバス選択手段は、前記2つの加算器の1つの少なくとも1つの入力について、前記複数の単方向入力データバスのうちのいずれか1つま

2

たは該加算器自身の出力のいずれかを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータまたは該加算器自身の出力のいずれかの一部を前記入力に与えるための手段を含む、請求項2に記載の並列処理のためのプロセッシングユニット。

【請求項6】 前記入力データバス選択手段が、前記2つの加算器の1つの少なくとも1つの入力について、前記複数の単方向入力データバスのうちのいずれか1つまたは前記乗算器の出力のいずれかを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータまたは前記乗算器の出力のいずれかの一部を前記入力に与えるための手段を含む、請求項2に記載の並列処理のためのプロセッシングユニット。

【請求項7】 前記入力データバス選択手段が、前記2つの加算器の1つの少なくとも1つの入力について、前記複数の単方向入力データバスのうちのいずれか1つまたは該加算器自身の出力または前記乗算器の出力の一部のいずれかを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータの一部または該加算器自身の出力または前記乗算器の出力の一部のいずれかを前記入力に与えるための手段を含む、請求項2に記載の並列処理のためのプロセッシングユニット。

【請求項8】 所定の情報を予め記憶するための読出専用記憶手段をさらに含み、

前記入力データバス選択手段が、前記乗算器の少なくとも1つの入力について、前記複数の単方向入力データバスのうちのいずれか1つまたは前記読出専用記憶手段の出力のいずれかを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータまたは前記記憶手段の出力のいずれかの一部を前記入力に与えるための手段を含む、請求項2に記載の並列処理のためのプロセッシングユニット。

【請求項9】 n 個のプロセッシングユニットと、隣り合うプロセッシングユニットを所定方向に円環状に順次に接続するための第1の単方向データバスと、

1つにおいて隣り合うプロセッシングユニットを円環状に順次に双方向に接続するための、第2の単方向データバスとを含み、

前記 n は4のべき乗であり、

各前記プロセッシングユニットは、

それぞれ複数個の入力を有し、与えられるデータの間に所定の演算を行なって結果を出力するための複数個の演算手段と、

隣接するプロセッシングユニットからの入力となる前記第1の単方向データバスと、前記第2の単方向データバスのうちの該プロセッシングユニットへの入力データバスとに接続され、前記複数個の演算手段の各入力ごとに前記第1および第2の単方向データバスのうちのいずれか1つを可制御的に選択して、選択された単方向データバスを介して与えられるデータの一部を前記入力に与え

3

るための入力データバス選択手段と、
前記複数の演算手段の出力に接続された入力と、隣接するプロセッシングユニットへの出力となる前記第1の単方向データバスと、前記第2の単方向データバスのうちの該プロセッシングユニットからの出力データバスとに接続された出力とを有し、前記演算手段の出力の各々を、前記第1および第2の単方向データバスのいずれかに出力するための出力データバス選択手段と、
前記複数の演算手段により所望の複合演算を実現するために、前記入力データバス選択手段と、前記出力データバス選択手段とによるデータの経路と、前記演算手段による演算の実行とを制御するための制御手段とを含む、並列処理プロセッサ。

【請求項10】 前記プロセッシングユニットと同数の、各々が一度に2つのデータを出力可能なデータ記憶手段と、

前記データ記憶手段の各々と、前記プロセッシングユニットの各々とを接続するための複数の読出データバスおよび複数の書込データバスとをさらに含み、

各前記プロセッシングユニットにおいて、

前記入力データバス選択手段は、隣接するプロセッシングユニットからの入力となる前記第1の単方向データバスと、前記第2の単方向データバスのうちの該プロセッシングユニットへの入力データバスと、前記複数の読出データバスとに接続され、前記複数の演算手段の各入力ごとに前記第1および第2の単方向データバスと前記読出データバスとのうちのいずれか1つを可制御的に選択して、選択されたデータバスを介して与えられるデータの一部を前記入力に与えるための手段を含み、

前記出力データバス選択手段は、前記複数の演算手段の出力に接続された入力と、隣接するプロセッシングユニットへの出力となる前記第1の単方向データバスと、前記第2の単方向データバスのうちの該プロセッシングユニットからの出力データバスと、前記書込データバスとに接続された出力とを有し、前記演算手段の出力の各々を、前記データバスのいずれかに出力するための手段とを含む、請求項9に記載の並列処理プロセッサ。

【請求項11】 前記プロセッシングユニットと同数の、各々が一度に2つのデータを出力可能なデータ記憶手段と、

前記データ記憶手段の各々と、前記プロセッシングユニットの各々とを接続するための複数の読出データバスおよび複数の書込データバスとをさらに含み、

前記複数のプロセッシングユニットは、各々が4の冪乗個のプロセッシングユニットを含む複数のグループに分割されており、

前記複数のプロセッシングユニットと前記複数のデータ記憶手段とは1対1に対応付けられており、

各前記プロセッシングユニットにおいて、

前記入力データバス選択手段は、隣接するプロセッシン

4

グユニットからの入力となる前記第1の単方向データバスと、前記第2の単方向データバスのうちの該プロセッシングユニットへの入力データバスと、前記複数の読出データバスのうちの該プロセッシングユニットが含まれるグループのプロセッシングユニットと対応付けられたデータ記憶手段からの読出データバスとに接続され、前記複数の演算手段の各入力ごとに前記第1および第2の単方向データバスと前記読出データバスとのうちのいずれか1つを可制御的に選択して、選択されたデータバスを介して与えられるデータの一部を前記入力に与えるための手段を含み、

前記出力データバス選択手段は、前記複数の演算手段の出力に接続された入力と、隣接するプロセッシングユニットへの出力となる前記第1の単方向データバスと、前記第2の単方向データバスのうちの該プロセッシングユニットからの出力データバスと、前記書込データバスのすべてとに接続された出力とを有し、前記演算手段の出力の各々を、前記データバスのいずれかに出力するための手段を含む、請求項9に記載の並列処理プロセッサ。

【請求項12】 4個のプロセッシングユニットと、隣り合うプロセッシングユニットを所定方向に円環状に順次に接続するための4本の第1の単方向データバスと、

1つにおいて隣り合うプロセッシングユニットを双方向に接続するための、4本の第2の単方向データバスとを含み、

各前記プロセッシングユニットは、

各々nビットの2つの入力とを有し、与えられるデータの間に乗算を行なって2nビット幅の結果を出力するための乗算手段と、

各々が、各々nビットの2つの入力とを有し、与えられるデータの間に加算を行なってnビット幅の結果を出力するための第1および第2の加算手段と、

前記第1の加算手段のキャリー出力を前記第2の加算手段のキャリー入力に可制御的に与えるためのキャリー切換手段と、

隣接するプロセッシングユニットからの入力となる前記第1の単方向データバスと、前記第2の単方向データバスのうちの該プロセッシングユニットへの入力データバスとに接続され、前記複数の演算手段の各入力ごとに前記第1および第2の単方向データバスのうちのいずれか1つを可制御的に選択して、選択された単方向データバスを介して与えられるデータの一部を前記乗算手段と前記加算手段との前記入力に与えるための入力データバス選択手段と、

前記乗算手段の出力と前記加算手段の出力とに接続された入力と、隣接するプロセッシングユニットへの出力となる前記第1の単方向データバスと、前記第2の単方向データバスのうちの該プロセッシングユニットからの出

5

カデータバスとに接続された出力とを有し、前記乗算手段および前記加算手段の出力の各々を、前記第1および第2の単方向データバスのいずれかに出力するための出力データバス選択手段と、

前記乗算手段および前記加算手段による所望の複合演算を実現するために、前記入力データバス選択手段と、前記出力データバス選択手段とによるデータの経路を制御するための制御手段と、

を含む並列処理プロセッサにおいて、所望の演算を行なうための動作方法であって、

演算に必要なデータの各々を前記入力データバス選択手段に与えるステップと、

前記入力データバス選択手段により、前記データの各々を上位および下位のnビットずつに分解し、前記4つのプロセッシングユニットの、前記乗算手段および前記加算手段の入力のいずれか2つにそれぞれ与えるステップと、

すべての前記プロセッシングユニットの前記キャリー切換手段を、前記所望の演算に応じて設定するステップと、

各前記プロセッシングユニットの前記出力データバス選択手段と前記入力データバス選択手段とを制御して、前記所望の演算が得られるように各前記プロセッシングユニットの前記乗算手段と、前記第1および第2の加算手段との間の接続を設定するステップと、

演算結果が、前記データバスのうちの所望のものに出力されるように、前記所望の演算によって定まる所定のプロセッシングユニットの前記乗算手段および加算手段のうちの所定のものの出力の、所定の部分を前記所望のデータバスに出力するように前記出力データバス選択手段を制御するステップとを含む、動作方法。

【請求項13】 前記接続を設定するステップは、あるプロセッシングユニット内の乗算手段または加算手段の出力の上位nビットが、他のプロセッシングユニット内の乗算手段または加算手段の下位nビットに入力されるように、これらプロセッシングユニットの前記出力データバス選択手段および前記入力データバス選択手段によるデータ経路を設定するステップを含む、請求項12に記載の動作方法。

【請求項14】 前記接続を設定するステップは、あるプロセッシングユニット内の乗算手段または加算手段の出力の上位nビットが、他のプロセッシングユニットへの出力データバスの下位nビットに出力されるように、該プロセッシングユニットの前記出力データバス選択手段によるデータ経路を設定するステップを含む、請求項12に記載の動作方法。

【請求項15】 各プロセッシングユニットごとに準備され、前記制御手段が実行する制御命令を格納するための複数の命令記憶手段をさらに含む、請求項9に記載の並列処理プロセッサ。

6

【請求項16】 前記制御命令は、対応のデータ記憶手段を制御するための第1の種類の制御命令と、前記制御手段による前記演算手段の制御のための第2の種類の制御命令との2つの系統に分類される、請求項15に記載の並列処理プロセッサ。

【請求項17】 前記第1の種類の制御命令は、前記データ記憶手段の各々の2つの読出アドレスと、1つの書込アドレスとを指定する、請求項16に記載の並列処理プロセッサ。

10 【請求項18】 前記制御手段は、与えられる第2の種類の制御命令が変更されるまでは、直前に与えられた第2の種類の制御命令に従って前記演算手段と前記入力データバス選択手段と前記出力データバス選択手段とを制御する、請求項16に記載の並列処理プロセッサ。

【請求項19】 前記命令記憶手段は、複数の命令を格納する命令メモリと、命令メモリの読出アドレスを指定するためのプログラムカウンタと、

前記プログラムカウンタにより指定されたアドレスを先頭として2つの命令を一度に読出するための手段と、

20 読出された2つの命令が同一の系統に属するか否かを判断するための手段と、

判断結果に従って、前記制御手段または前記データ記憶手段またはその双方に制御命令を与えるための手段と、

判断結果に従って、前記プログラムカウンタのカウントを1または2増加させるための手段とを含む、請求項16に記載の並列処理プロセッサ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】この発明は、複数のプロセッシングユニットにより構成される並列処理プロセッサの改良に関し、特に、幅広い演算に対応できる、制御の容易な並列処理プロセッサと、そのためのプロセッシングユニットと、並列処理プロセッサの動作方法とに関する。

【0002】

【従来の技術】従来の並列処理プロセッサ（以下単に「プロセッサ」と呼ぶ）の構成を図16および図17に示す。このプロセッサは、「ISSCCダイジェスト・オブ・テクニカル・ペーパーズ」（“ISSCC Digest of Technical Papers”、1991年2月、252～253頁）に発表されたプロセッサのブロック構成のうち、本願発明に関連する主要演算部分を抜粋し、簡略化して示したものである。

【0003】図16を参照して、このプロセッサは、4個のプロセッシングユニットPU00～PU11（図16中では符号30、32、34、36により示される）と、アドレス演算ユニット（AU）48と、ワーキングメモリ38と、データキャッシュメモリ40、42、44、46とを含む。各プロセッシングユニットからデータキャッシュメモリ40、42、44、46およびワー

7

キングメモリ38へのアクセスは、4本のキャッシュメモリ読出専用バス50と、6本の読出書込兼用バス42との、合計10本のバスを介して行なわれる。このプロセッサはさらに、レジスタファイル58と、セクタ56とを含む。セクタ56とレジスタファイル58とは、プロセッシングユニット間のデータ交換を行なうためのものであり、キャッシュメモリ読出専用バス50と、読出書込兼用バス52と、SBUS54とから読込んだデータをレジスタファイル58を介して各プロセッシングユニット30、32、34、36とバス52、54とに出力可能である。

【0004】図17を参照して、各プロセッシングユニット30、32、34、36は、類似ではあるが相互にやや異なった構成となっている。各プロセッシングユニット30、32、34、36は、演算器ALU70、72、74、76と、乗算器MPY80、82、84、86と、加算器ADD90、92、94、96とを含む。また各プロセッシングユニット30、32、34、36内には、ALU、乗算器、加算器への入力を選択するためのセクタが含まれている。これについては後述するが、図17においては、図面の簡略化のために、セクタの入出力については簡略化して示してある。

【0005】図17を参照して、たとえばプロセッシングユニット30は、5:1のセクタ110と112とを含む。プロセッシングユニット30はさらに、ALU70の出力の一方と、レジスタファイル58からの出力と、セクタ110の出力とから1つを選択してALU70の一方の入力に与えるためのセクタ130と、セクタ112の出力と、後述するプロセッシングユニット32から与えられるデータとの一方を選択してALU70の他方の入力に与えるためのセクタ132とを含む。プロセッシングユニット30はさらに、ALU70の出力の一方とレジスタファイル58の出力とのいずれか一方を選択してMPY80の一方の入力に与えるためのセクタ150と、プロセッシングユニット32から与えられるデータと、ADD90の出力とのいずれか一方を選択してADD90の一方の入力に与えるためのセクタ162とを含む。プロセッシングユニット30はさらにセクタ160を含んでおり、このセクタ160はMPY80の出力とADD90の出力とのいずれか一方を選択してデータバス52に出力するためのものである。

【0006】プロセッシングユニット32は、同様にセクタ114、116、134、136、152、164を含む。プロセッシングユニット34は、セクタ118、120、138、140、154、166、168を含む。プロセッシングユニット36は、セクタ122、124、142、144、156、170を含む。さらに、セクタ160と同様のセクタが各プロセッシングユニット32、34、36に含まれている

8

が、図の簡略化のため図17には示していない。

【0007】セクタ114、118、122は、セクタ110と同様の機能を有する。セクタ116、120、124は、セクタ112と同様の機能を有する。セクタ134、138、142は、セクタ130と同様の機能を有する。セクタ136は、セクタ116の出力とセクタ112の出力との一方を選択してALU72に与えるためのものである。セクタ140はセクタ132と同様である。セクタ144は、セクタ124の出力とセクタ120の出力との一方をALU76に与えるためのものである。セクタ152、154、156は、セクタ150と同様の機能を有する。セクタ164は、プロセッシングユニット34の出力と乗算器82の出力とのいずれか一方を選択して加算器92に与えるためのものである。セクタ166は、プロセッシングユニット30の出力と、加算器94の出力とのいずれか一方を選択して加算器94に与えるためのものである。セクタ168は、乗算器84の出力と、プロセッシングユニット36の出力とのいずれか一方を選択して加算器94に与えるためのものである。セクタ170は、乗算器84の出力と、加算器96の出力とのいずれか一方を選択して加算器96に与えるためのものである。

【0008】各プロセッシングユニット30、32、34、36には、各プロセッシングユニットを制御するためのローカル命令メモリLPM00、01、10、11（図17中では参照符号100、102、104、106で示される）が設けられている。

【0009】アドレス演算ユニット48は、各メモリ38、40、42、44、46の読出、書込アドレスを演算するためのものである。

【0010】図16および図17に示されるごとく、従来のプロセッサにおいては、プロセッシングユニットの構成は相互に異なっており、相互の間の接続も、処理対象となる演算に合わせて特殊な形態となっている。

【0011】図18および図19を参照して、従来のプロセッサは次のように動作する。プロセッシングユニット間のバス接続は、図18と図19とに示される2種類の構成から選択することができる。図18に示される例においては、プロセッシングユニット30、32、34、36の間でのデータバス接続が存在しないように各セクタが設定される。各プロセッシングユニットでは、積和演算が行なわれる。

【0012】図19に示される例では、プロセッシングユニット32の乗算器82の出力がプロセッシングユニット30の加算器90の入力に与えられる。加算器90の出力は、プロセッシングユニット34の加算器94の入力に与えられる。一方プロセッシングユニット34の乗算器84の出力がプロセッシングユニット36の加算器96の入力の一方に与えられる。加算器96の出力は

プロセッシングユニット34の加算器194の入力の他方に与えられる。加算器94の出力はプロセッシングユニット32の加算器92の入力の一方に与えられる。この図19に示される接続では、4項ごとの積和演算が可能である。4項ごとの積和結果はプロセッシングユニット32の出力として得られる。

【0013】このプロセッサの制御は、セットアップ命令1つと、各プロセッシングユニットの制御を行なうための4個の命令との、合計5個の命令を単位として行なわれる。各命令は32ビットであり、5個の命令では160ビットとなる。

【0014】セットアップ命令は、各プロセッシングユニット入力部の5:1セレクタ110、112、114、116、118、120、122、124や、各プロセッシングユニット間のデータバスの接続を設定するためのセレクタなどを制御する。プロセッシングユニット制御命令は、メモリ38、40、42、44、46のアドレスを発生したり、ローカル命令メモリ100、102、104、106のアドレス指定を行なったりする。ローカル命令メモリ100、102、104、106に含まれるローカル命令は、演算器で行なう演算内容を指定するためのものである。

【0015】

【発明が解決しようとする課題】図16～図19に示される従来技術のプロセッサには、次のような問題点がある。このプロセッサは、もともと動画像圧縮用に積和演算の効率化を目標として開発された。そのため、同じように大量の演算が要求される処理であっても、動画像圧縮以外の分野へこのプロセッサを適用することは困難である。データの大量処理が要求される演算としては、積和演算のほかにもFFT（高速フーリエ変換）に用いられるバタフライ演算や、科学技術計算における倍精度演算などがある。バタフライ演算には、乗算器4個と加算器6個とが必要である。倍精度乗算（2nビットとする）では、 $n \times n$ ビットの乗算器4個と $2n + 2n$ ビットの加算器3個とが必要である。しかし、図16～図19に示されるプロセッサが行なえる処理は、動画像圧縮用の処理だけであり、そのハードウェアも、アルゴリズムが固定したものとして実現されている。これは、複数のプロセッシングユニットを備えたプロセッサにおいて、上述のような様々な処理を行なおうとするとその制御が複雑になるなどの理由によるものである。したがって従来のこの種のプロセッサで汎用できるものは極めて少数であり、しかもその制御が複雑であったり、ハードウェアが複雑であるという欠点がある。

【0016】この発明は上述の問題点に鑑みてなされたものであって、複数のプロセッシングユニットを備えることにより並列処理を効率よく行なえるとともに、幅広い種類の演算を、比較的単純な制御方法で可能とする並列処理プロセッサとそのためのプロセッシングユニッ

トと、プロセッサの動作方法とを提供することを目的とする。

【0017】

【課題を解決するための手段】請求項1に記載の並列処理のためのプロセッシングユニットは、それぞれ複数の入力を有し、与えられるデータの間に所定の演算を行なって結果を出力するための複数の演算手段と、複数の単方向入力データバスに接続され、複数の演算手段の各入力ごとに、複数の単方向入力データバスのうちのいずれか1つを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータの一部を演算手段の入力に与えるための入力データバス選択手段と、複数の演算手段の出力に接続された入力と、単方向入力データバスと同じ数の単方向出力データバスに接続された出力とを有し、演算手段の出力を、単方向出力データバスのいずれかに出力するための出力データバス選択手段と、複数の演算手段により所望の複合演算を実現するために、入力データバス選択手段と、出力データバス選択手段とによるデータの経路を制御するための制御手段とを含む。

【0018】請求項2に記載のプロセッシングユニットは、請求項1に記載のものであって、その複数の演算手段が、2つのnビット幅の入力を有し、与えられる2つのデータを乗算して2nビットの結果を出力する乗算器と、各々が2つのnビット幅の入力を有し、与えられる2つのデータを加算してnビット幅の結果を出力する2つの加算器とを含む。

【0019】請求項3に記載のプロセッシングユニットは、請求項2に記載のものであって、複数の単方向入力データバスおよび複数の単方向出力データバスの各々は2nビット幅を有し、入力データバス選択手段は、複数の演算手段の各入力ごとに複数の単方向入力データバスのうちのいずれか1つを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータの上位または下位のnビットを該入力に与えるための手段を含み、出力データバス選択手段は、演算手段の出力の各々を、単方向出力データバスの任意のいずれかの上位nビットまたは下位nビットまたはその双方に出力可能とするための手段を含む。

【0020】請求項4に記載のプロセッシングユニットは、請求項2に記載のものであって、2つの加算器の一方はキャリー出力を有し、他方はキャリー入力を有し、さらに、キャリー出力とキャリー入力とを可制御的に断続するための手段を含む。

【0021】請求項5に記載のプロセッシングユニットは請求項2に記載のものであって、入力データバス選択手段が、2つの加算器の1つの少なくとも1つの入力について、単方向入力データバスのうちのいずれか1つまたは該加算器自身の出力のいずれかを可制御的に選択して、選択された単方向入力データバスを介して与えられ

11

るデータまたは該加算器自身の出力のいずれかの一部を該入力に与えるための手段を含む。

【0022】請求項6に記載のプロセッシングユニットは、請求項2に記載のものであって、入力データバス選択手段が、2つの加算器の1つの少なくとも1つの入力について、複数の単方向入力データバスのうちのいずれか1つまたは乗算器の出力のいずれかを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータまたは乗算器の出力のいずれかの一部を該入力に与えるための手段を含む。

【0023】請求項7に記載のプロセッシングユニットは、請求項2に記載のものであって、入力データバス選択手段が、2つの加算器の1つの少なくとも1つの入力について、複数の単方向入力データバスのうちのいずれか1つまたは該加算器自身の出力または乗算器の出力の一部のいずれかを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータの一部または該加算器自身の出力または乗算器の出力の一部のいずれかを該入力に与えるための手段を含む。

【0024】請求項8に記載のプロセッシングユニットは、請求項2に記載のものであって、所定の情報を予め記憶するための読出専用記憶手段をさらに含む。入力データバス選択手段は、乗算器の少なくとも1つの入力について、単方向入力データバスのうちのいずれか1つまたは読出専用記憶手段の出力のいずれかを可制御的に選択して、選択された単方向入力データバスを介して与えられるデータまたは読出専用記憶手段の出力のいずれかの一部を該入力に与えるための手段を含む。

【0025】請求項9に記載の並列処理プロセッサは、 n 個のプロセッシングユニットと、隣り合うプロセッシングユニットを所定方向に円環状に順次に接続するための第1の単方向データバスと、1つにおいて隣り合うプロセッシングユニットを円環状に順次に双方向に接続するための、第2の単方向データバスとを含む。 n は4のべき乗である。各プロセッシングユニットは、それぞれ複数の入力を有し、与えられるデータの間に所定の演算を行なって結果を出力するための複数の演算手段と、隣接するプロセッシングユニットからの入力となる第1の単方向データバスと、第2の単方向データバスのうちの該プロセッシングユニットへの入力データバスとに接続され、複数の演算手段の各入力ごとに第1および第2の単方向データバスのうちのいずれか1つを可制御的に選択して、選択された単方向データバスを介して与えられるデータの一部を該入力に与えるための入力データバス選択手段と、複数の演算手段の出力に接続された入力と、隣接するプロセッシングユニットへの出力となる第1の単方向データバスと、第2の単方向データバスのうちの該プロセッシングユニットからの出力データバスとに接続された出力とを有し、演算手段の出力の各々を、第1および第2の単方向データバスのいずれかに出

12

力するための出力データバス選択手段と、複数の演算手段により所望の複合演算を実現するために、入力データバス選択手段と、出力データバス選択手段とによるデータの経路と、演算手段による演算の実行とを制御するための制御手段とを含む。

【0026】請求項10に記載の並列処理プロセッサは、請求項9に記載のものであって、プロセッシングユニットと同数の、各々が一度に2つのデータを出力可能なデータ記憶手段と、データ記憶手段の各々と、プロセッシングユニットの各々とを接続するための複数の読出データバスと複数の書込データバスとをさらに含む。各プロセッシングユニットにおいて、入力データバス選択手段は、隣接するプロセッシングユニットからの入力となる第1の単方向データバスと、第2の単方向データバスのうちの該プロセッシングユニットへの入力データバスと、複数の読出データバスとに接続され、複数の演算手段の各入力ごとに第1および第2の単方向データバスと読出データバスとのうちのいずれか1つを可制御的に選択して、選択されたデータバスを介して与えられるデータの一部を入力に与えるための手段を含む。出力データバス選択手段は、複数の演算手段の出力に接続された入力と、隣接するプロセッシングユニットへの出力となる第1の単方向データバスと、第2の単方向データバスのうちの該プロセッシングユニットからの出力データバスと、書込データバスとに接続された出力とを有し、演算手段の出力をこれらデータバスのいずれかに出力するための手段とを含む。

【0027】請求項11に記載のプロセッサは、請求項9に記載のものであって、プロセッシングユニットと同数の、各々が一度に2つのデータを出力可能なデータ記憶手段と、データ記憶手段の各々と、プロセッシングユニットの各々とを接続するための複数の読出データバスと複数の書込データバスとをさらに含む。複数のプロセッシングユニットは、各々が4のべき乗個のプロセッシングユニットを含む複数のグループに分割されており、複数のプロセッシングユニットと複数のデータ記憶手段とは1対1に対応付けられている。各プロセッシングユニットにおいて、入力データバス選択手段は、隣接するプロセッシングユニットからの入力となる第1の単方向データバスと、第2の単方向データバスのうちの該プロセッシングユニットへの入力データバスと、複数のデータバスのうち該プロセッシングユニットが含まれるグループのプロセッシングユニットと対応付けられたデータ記憶手段からの読出データバスとに接続され、複数の演算手段の各入力ごとに第1および第2の単方向データバスと読出データバスとのうちのいずれか1つを可制御的に選択して、選択されたデータバスを介して与えられるデータの一部を該入力に与えるための手段を含む。出力データバス選択手段は、複数の演算手段の出力に接続された入力と、隣接するプロセッシングユニッ

13

トへの出力となる第1の単方向データバスと、第2の単方向データバスのうちの該プロセッシングユニットからの出力データバスと、書込データバスのすべてに接続された出力とを有し、演算手段の出力を、データバスのいずれかに出力するための手段を含む。

【0028】請求項12に記載の並列処理プロセッサの動作方法は、4個のプロセッシングユニットと、隣り合うプロセッシングユニットを所定方向に円環状に順次に接続するための4本の第1の単方向データバスと、1つにおいて隣り合うプロセッシングユニットを双方向に接続するための、4本の第2の単方向データバスを含む並列処理プロセッサの動作方法である。各プロセッシングユニットは、各々nビットの2つの入力を受け、与えられるデータの間に乗算を行なって2nビット幅の結果を出力するための乗算手段と、各々が、各々nビットの2つの入力を受け、与えられるデータの間に加算を行なってnビット幅の結果を出力するための第1および第2の加算手段と、第1の加算手段のキャリー出力を第2の加算手段のキャリー入力に可制御的に与えるためのキャリー切替手段と、隣接するプロセッシングユニットからの入力となる第1の単方向データバスと、第2の単方向データバスのうちの該プロセッシングユニットへの入力データバスとなるものとに接続され、乗算手段と加算手段との各入力ごとに第1および第2の単方向データバスのうちのいずれか1つを可制御的に選択して、選択された単方向データバスを介して与えられるデータの一部を乗算手段と加算手段との入力にそれぞれ与えるための入力データバス選択手段と、乗算手段および加算手段の出力に接続された入力と、隣接するプロセッシングユニットへの出力となる第1の単方向データバスと、第2の単方向データバスのうちの該プロセッシングユニットからの出力データバスとに接続された出力とを有し、乗算手段および加算手段の出力を、第1および第2の単方向データバスのいずれかに出力するための出力データバス選択手段と、乗算手段および加算手段により所望の複合演算を実現するために、入力データバス選択手段と、出力データバス選択手段とによるデータの経路を制御するための制御手段とを含む。この動作方法は演算に必要なデータの各々を入力データバス選択手段に与えるステップと、入力データバス選択手段により、データの各々を上位および下位のnビットずつに分解し、4つのプロセッシングユニットの、乗算手段および加算手段の入力のいずれか2つにそれぞれ与えるステップと、すべてのプロセッシングユニットのキャリー切替手段を、所望の演算に応じて設定するステップと、各プロセッシングユニットの出力データバス選択手段および入力データバス選択手段を制御して、所望の演算が得られるように各プロセッシングユニットの乗算手段と、第1および第2の加算手段との間の接続を設定するステップと、演算結果が、データバスのうちの所望のものに出力されるように、所

14

望の演算によって定まる所定のプロセッシングユニットの乗算手段および加算手段のうちの所定のものの出力の、所定の部分を所望のデータバスに出力するように出力データバス選択手段を制御するステップとを含む。

【0029】請求項13に記載の動作方法は、請求項12に記載のものであって、接続を設定するステップは、あるプロセッシングユニット内の乗算手段または加算手段の出力の上位nビットが、他のプロセッシングユニット内の乗算手段または加算手段の下位nビットに入力されるように、これらプロセッシングユニットの出力データバス選択手段および入力データバス選択手段によるデータ経路を設定するステップを含む。

【0030】請求項14に記載の動作方法は、請求項12に記載のものであって、接続を設定するステップは、あるプロセッシングユニット内の乗算手段または加算手段の出力の上位nビットが、他のプロセッシングユニットへのデータバスの下位nビットに出力されるように、該プロセッシングユニットの出力データバス選択手段によるデータ経路を設定するステップを含む。

【0031】請求項15に記載の並列処理プロセッサ、請求項9に記載のものであって、各プロセッシングユニットごとに準備された、制御手段が実行する制御命令を格納するための命令記憶手段をさらに含む。

【0032】請求項16に記載の並列処理プロセッサは、請求項15に記載のものであって、制御命令は、データ記憶手段を制御するための第1の種類の制御命令と、制御手段による演算手段の制御のための第2の種類の制御命令との2つの系統に分類される。

【0033】請求項17に記載の並列処理プロセッサは、請求項16に記載のものであって、第1の種類の制御命令は、データ記憶手段の各々の2つの読出アドレスと、1つの書込アドレスとを指定する。

【0034】請求項18に記載の並列処理プロセッサは、請求項16に記載のものであって、制御手段は、与えられる第2の種類の制御命令が変更されるまでは、直前に与えられた第2の種類の制御命令に従って演算手段と入力データバス選択手段と出力データバス選択手段とを制御する。

【0035】請求項19に記載の並列処理プロセッサは、請求項16に記載のものであって、命令記憶手段は、複数個の命令を記憶する命令メモリと、命令メモリの読出アドレスを指定するためのプログラムカウンタと、プログラムカウンタにより指定されたアドレスを先頭として2つの命令を一度に読出するための手段と、読出された2つの命令が同一の系統に属するか否かを判断するための手段と、判断結果に従って、制御手段またはデータ記憶手段またはその双方に制御命令を与えるための手段と、判断結果に従って、プログラムカウンタのカウントを1または2増加させるための手段とを含む。

【0036】

15

【作用】請求項1に記載の並列処理のためのプロセッシングユニットにおいては、そこに接続される単方向入力データバスの数と、単方向出力データバスの数とが同じである。そしてこれら単方向入力データバスの任意のものを介して与えられるデータを、複数個の演算手段で処理して結果を単方向出力データバスの任意のものに出力できる。制御手段の制御により種々の演算が可能のため、このプロセッシングユニットは汎用的であり、かつ制御手段の制御は比較的単純でよい。しかも入出力データバスの数が同一であるために、このプロセッシングユニットを複数個組合わせて並列処理プロセッサを作製するときのプロセッシングユニット相互の接続が容易であり、各プロセッシングユニット構成が同一であるために、それらの制御に互換性を持たせることができる。

【0037】請求項2に記載のプロセッシングユニットは、 $n \times n$ ビットの乗算器と $n + n$ ビットの2つの加算器とを用いて、大量のデータ処理が要求される演算における典型的な処理を行なうことができる。

【0038】請求項3に記載のプロセッシングユニットでは、演算手段の各入力に、単方向入力データバスのうちのいずれか任意の1つの上位または下位の n ビットが与えられる。これら演算手段の出力の各々は、単方向出力データバスの任意のいずれかの上位 n ビットまたは下位 n ビットまたはその双方に出力される。したがって $2n$ ビットのデータ同士の演算を、それぞれ上位 n ビットと下位 n ビットとに分けて $n \times n$ ビットの乗算器および $n + n$ ビットの加算器を用いて行なうことができる。

【0039】請求項4に記載のプロセッシングユニットでは、2つの加算器の一方のキャリー出力を他方のキャリー入力とする場合と、しない場合とを選択できる。したがって、 $2n$ ビットのデータ同士の加算と、 n ビットのデータ同士の加算と、 n ビットのデータ同士の2つの別個の加算とを実行することができる。

【0040】請求項5に記載のプロセッシングユニットでは、2つの加算器の1つの少なくとも1つの入力について、入力データとして複数の単方向入力データバスのうちのいずれか1つまたは該加算器自身の出力のいずれかを選択できるので、自己の加算結果を用いる演算、たとえば積和演算と、データバス経由のデータを用いた加算との双方を行なうことができる。

【0041】請求項6に記載のプロセッシングユニットでは、2つの加算器の1つの少なくとも1つの入力について、入力データとして単方向入力データバスのうちのいずれか1つまたは乗算器の出力のいずれかを選択できるので、乗算結果を加算の入力に用いる演算、たとえば積和演算と、データバス経由のデータを用いた演算との双方を行なうことができる。

【0042】請求項7に記載のプロセッシングユニットでは、2つの加算器の1つの少なくとも1つの入力について、入力データとして単方向入力データバスのうちの

16

いずれか1つと、該加算器自身の出力と、乗算器の出力の一部とのいずれかを選択できる。したがって、乗算結果や加算結果を加算の入力に用いる演算、たとえば積和演算と、データバス経由のデータを用いた加算との双方を行なうことができる。

【0043】請求項8に記載のプロセッシングユニットでは、乗算器の少なくとも1つの入力について、入力データとして単方向入力データバスのうちのいずれか1つと読出専用記憶手段の出力とのいずれかを選択できる。したがって、読出専用記憶手段に格納されたデータを用いることにより処理の高速化を図れる処理、たとえばニュートン・ラプソン法による除算や、開平演算などを効率よく行なうための構成と、通常の演算を行なうための構成とを自由に選択できる。

【0044】請求項9に記載の並列処理プロセッサでは、 n 個のプロセッシングユニットが、隣り合うプロセッシングユニットとは第1の単方向データバスにより単方向に、1つおいて隣り合うプロセッシングユニットとは第2の単方向データバスにより双方向に、それぞれ接続される。各プロセッシングユニットは、最低でも2以上の同じ数の入力と出力とを有する。入出力の数が多く、演算の自由度が高くなる。また、各プロセッシングユニットは同一の構造を有するため、プロセッサのレイアウトおよび制御の複雑さが低減される。

【0045】請求項10に記載の並列処理プロセッサでは、各プロセッシングユニットごとに、データ記憶手段が準備される。各プロセッシングユニットは、データ記憶手段からのデータと、他のプロセッサからのデータとのいずれにも、複数個の演算手段を用いた所定の処理を行ない、任意の演算手段の出力を他のプロセッシングユニットと、データ記憶手段との任意のものに出力できる。

【0046】請求項11に記載の並列処理プロセッサでは、プロセッシングユニットはグループに分類され、各プロセッシングユニットごとに、データ記憶手段が準備される。各プロセッシングユニットは、同一のグループに属するプロセッシングユニットに対応するデータ記憶手段からのデータと、他のプロセッサからのデータとのいずれにも、複数個の演算手段を用いた所定の処理を行ない、任意の演算手段の出力を他のプロセッシングユニットと、データ記憶手段との任意のものに出力できる。データ記憶手段から、同じグループに属するプロセッシングユニットへの読出データバスは、プロセッシングユニットをグループ化せず全体に接続する場合と比較して短くてすみ、プロセッシングユニットへのデータの供給が高速に行なえる。また1グループのプロセッシングユニットで共通のデータを用いた演算を実行できる。さらに、各プロセッシングユニットからは任意のデータ記憶手段にデータを蓄積めるので、グループ間のデータ交換も可能である。

17

【0047】請求項12に記載の並列処理プロセッサの動作方法では、データの各々が上位および下位のnビットずつに分解され、4つのプロセッシングユニットの、乗算手段および加算手段の入力のいずれか2つにそれぞれ与えられる。他のプロセッシングユニットの演算結果の上位nビットまたは下位nビットは、新たな演算のためのデータの上位nビットまたは下位nビットとして乗算手段または加算手段に与えることができ、多様な演算を行なうことができる。また、2つの加算手段の間でキャリーの入出力を行なうかどうかを選択できるので、n

ビット精度の2つの加算処理と2nビット精度の1つの加算処理とを切替えて行なうことができる。したがって、非常に自由度の高い処理を行なうことができる。

【0048】請求項13に記載の並列処理プロセッサの動作方法では、あるプロセッシングユニット内の乗算手段の出力の上位nビットまたは下位のnビットが、他のプロセッシングユニット内の乗算手段または加算手段の下位nビットと上位nビットとの任意の一方に入力されるようにデータバスと乗算手段または加算手段との接続を設定できる。シフト手段を用いることなく多様な演算を行なうことが可能である。

【0049】請求項14に記載の並列処理プロセッサの動作方法では、あるプロセッシングユニット内の乗算手段の出力の上位nビットが、他のプロセッシングユニット内の乗算手段または加算手段の下位nビットに入力されるようにデータバスと乗算手段または加算手段との接続を設定できる。シフト手段を用いることなくデータを実質的にシフトでき、シフトのための時間も必要としない。

【0050】請求項15に記載の並列処理プロセッサでは、各プロセッシングユニットごとに制御命令が命令記憶手段に格納される。複雑な演算を各プロセッシングユニットごとに制御すればよく、並列処理プロセッサ全体の制御が容易になる。

【0051】請求項16に記載の並列処理プロセッサでは、制御命令は、データ記憶手段を制御するための制御命令と、演算手段の制御のための制御命令との2つの系統に分類される。データ記憶手段から出力されるデータを介しつつ一定の命令を繰り返し行なう場合などに、データ記憶手段を制御する制御命令のみを、その内容を変化させつつ繰り返し発行すれば、演算手段のための異なる制御命令をいくつも発行する必要がない。

【0052】請求項17に記載の並列処理プロセッサでは、データ記憶手段の各々の2つの読出アドレスからの2つのデータを読出して所定の演算を行ない、結果を1つの書込アドレスにより指定された位置に書込できる。

【0053】請求項18に記載の並列処理プロセッサでは、第2の種類の制御命令が変更されるまでは、直前に与えられた第2の種類の制御命令に従って演算手段と入力データバス選択手段と出力データバス選択手段とが制

18

御される。データ記憶手段から出力されるデータを介しつつ一定の命令を繰り返し行なう場合などに、データ記憶手段を制御する命令のみを、その内容を変化させつつ繰り返し発行すればよく、演算手段のための同一の制御命令を繰り返し発行する必要がない。

【0054】請求項19に記載の並列処理プロセッサでは、命令メモリから、プログラムカウンタにより指定されたアドレスを先頭として2つの命令が一度に読出され、読出された2つの命令が同一の系統に属するか否かが判断される。同一であれば先に読出された命令のみ実行し、プログラムカウンタを1増加させる。同一でなければ制御手段とデータ記憶手段との双方にそれぞれの制御命令を与え、プログラムカウンタを2増加させる。異なる系統の命令は同時に実行可能なので、一度に制御命令を1つしか読出さない場合と比較して、命令の実行速度が向上する。

【0055】

【実施例】

【第1の実施例】本発明の一実施例を、4個のプロセッシングユニット(PU)から構成されるプロセッサについて以下に示す。図1を参照してこのプロセッサは、4つのプロセッシングユニットPU00、01、10、11(図1中ではそれぞれ参照符号200、202、204、206により示される)と、これらプロセッシングユニット200、202、204、206を、この順序で円環状に接続するための単方向データバス210、212、214、216を含む。プロセッシングユニットPUに付けられた数値(00、01、10、11)は、そのプロセッシングユニットのアドレスを2進数で表わしたものである。

【0056】このプロセッサはさらに、アドレスが2だけ異なるプロセッシングユニットを双方向に接続するための、単方向入力データバス220、222、224、226を含む。前述の単方向データバス210、212、214は、アドレスが1だけ異なるプロセッシングユニットを順次単方向に接続するためのものである。またデータバス216は、プロセッシングユニットPU11(206)からPU00(200)へのデータバスである。図1において、各データバスに付加された矢印は、データの流れる方向を表わしている。図1に示されるプロセッサでは、各プロセッシングユニットに入るデータバスの入力数と、各プロセッシングユニットから出るデータバスの出力数とは相互に等しく、それぞれ2となっている。すべてのプロセッシングユニットにおいてこの数は等しい。

【0057】図2を参照して、この第1の実施例のプロセッサは、前述のプロセッシングユニットPU00、01、10、11(200、202、204、206)と、データバス210、212、214、216、220、222、224、226とに加えて、各プロセッサ

19

200、202、204、206に対応してそれぞれ設けられたコントロール回路(PUC)250、252、254、256と、命令メモリ(IM)260、262、264、266と、データメモリ270、272、274、276を含む。

【0058】このプロセッサはさらに、データメモリ270、272、274、276から読出されるデータのためのメモリ読出バス群280と、各プロセッシングユニット200、202、204、206からデータメモリ270、272、274、276に書込むデータのためのメモリ書込バス群272を含む。各データメモリ270、272、274、276はそれぞれ同時に2個のデータの読出が可能であり、それぞれデータバス300と301、302と303、304と305、306と307によりメモリ読出バス群280内の互いに異なるデータバスに接続されている。この接続については後述する。メモリ読出データバス群280内のデータバスは、データバス群310、312、314、316によりプロセッシングユニット200、202、204、206にそれぞれ接続される。

【0059】メモリ読出データバス群282は、4つのnビットデータバスを含み、同じくそれぞれ4本ずつのnビット幅のデータバスからなるデータバス群320、322、324、326によりプロセッシングユニット200、202、204、206の出力に接続されている。この接続の詳細については後述する。データメモリ270、272、274、276へのデータの書込は同時に1個が可能である。データメモリに書込むデータは各プロセッシングユニットの演算結果である。

【0060】命令メモリ260、262、264、266へは、図示されない入出力ポートを介して外部より与えられる命令が格納される。コントロール回路250、252、254、256の各々は、2個の出力を持ち、出力の一方はデータメモリ270、272、274、276の制御に、他方はプロセッシングユニット200、202、204、206内の演算器制御にそれぞれ用いる。

【0061】図3に、プロセッシングユニット200の内部構成を示す。図3にはプロセッシングユニット200の構成を例として示すが、他のプロセッシングユニット202、204、206の構成もこのプロセッシングユニット200と全く同一である。したがって、ここではそれらについての詳しい説明は繰り返さないこととする。

【0062】図3を参照して、メモリ書込バス群282は、4本のメモリ書込バス400、402、404、406を含む。また図1および図2に示されるプロセッシングユニット202へのデータバス210は、上位nビットのデータバス210Mと下位nビットのデータバス210Lを含む。同様にプロセッシングユニット20

20

4へのデータバス220は、上位nビットのデータバス220Mと下位nビットのデータバス220Lを含む。

【0063】また図2に示されるメモリ読出バス群280からプロセッシングユニット200へのメモリ読出バス群310は、図3に示されるようにデータバス330、332、334、336、340、342、344、346を含む。前述のようにメモリ読出バス群280は8本のデータバスを含み、これらメモリ読出バス330、332、334、336、340、342、344、346はその8本のデータバスからそれぞれ分岐したものである。この接続については図4を参照して後述する。

【0064】図1および図2に示されるプロセッシングユニット204からのデータバス224は、上位nビットのデータバス224Mと下位nビットのデータバス224Lを含む。またプロセッシングユニット206からのデータバス216も同様に、上位nビットのデータバス216Mと下位nビットのデータバス216Lを含む。

【0065】図3を参照して、このプロセッシングユニット200は、読出バス群310に接続されたセクタ350、352、354、356と、セクタ350、352、354、356の出力とデータバス216、224となどが入力に接続されたセクタ360、362、364、366、368、370と、セクタ360と362との出力が入力に接続された乗算器(MPY)380と、セクタ368と370との出力に接続された2つの入力を有する第1の加算器(ADD0)384と、セクタ364と366との出力に接続された2つの入力を有する第2の加算器(ADD1)382と、乗算器380のそれぞれnビット幅の出力MPMとMPLと、加算器384、382の出力とに接続され、各出力データを出力データバス210、220の上位nビットのデータバス210Mおよび220Mと、下位nビットのデータバス210Lおよび220Lとのいずれかに出力するためのクロスバスイッチ(CBS)392と、乗算器380の2つの出力MPM、MPLと、2つの加算器382、384の出力とを、メモリ書込バス群282の4つのメモリ書込バス400、402、404、406のいずれかに出力するためのセクタ(SELW)390を含む。セクタ390とメモリ書込バス400、402、404、406とはそれぞれデータバス410、412、414、416により接続されている。クロスバスイッチ392は、それぞれ2nビットのデータバス420、422によりデータバス210、220に接続されている。データバス420の上位nビットがデータバス210Mに接続され、下位nビットがデータバス210Lに接続されている。データバス422の上位nビットがデータバス220Mに接続さ

21

れ、下位nビットがデータバス220Lに接続されている。

【0066】セクタ350および354の入力には、4組のデータバス330、332、334、336がそれぞれ接続されている。セクタ352、356の入力には、4組のデータバス340、342、344、346がそれぞれ接続されている。

【0067】セクタ360の一方の入力にはセクタ350の出力が接続されている。セクタ360の他方の入力には、データバス224Mの上位nビットのデータバス224Mが接続されている。セクタ362の一方の入力にはセクタ352の出力が接続されている。セクタ362の他方の入力には、データバス216Mの上位nビット216Mが接続されている。

【0068】セクタ364、366はそれぞれ3入力である。セクタ364の1つの入力には、セクタ354の出力が接続されている。セクタ364の他の1つの入力には、データバス224Mの上位nビットのデータバス224Mが接続されている。セクタ364の残りの1つの入力には、加算器382の出力AD1が接続されている。セクタ366の入力の1つにはセクタ356の出力が接続されている。セクタ366の他の入力の1つには、乗算器380の出力のうちの上位nビットMPMが与えられる。セクタ366の残りの1つの入力には、データバス216Mの上位nビットのデータバス216Mが接続される。

【0069】セクタ368、370はそれぞれ4入力を有する。セクタ368の第1の入力には、加算器384の出力AD0が与えられる。第2の入力には、セクタ350の出力が与えられる。第3の入力にはデータバス224の下位nビットのデータバス224Lが接続される。第4の入力には、データバス224Mの上位nビットのデータバス224Mが接続される。セクタ370の第1の入力は、セクタ352の出力に接続される。第2の入力はデータバス216の下位nビットのデータバス216Lに接続される。第3の入力には乗算器MPY380の出力の下位nビットMPLが与えられる。第4の入力はセクタ356の出力に接続される。

【0070】加算器384のキャリー出力と加算器382のキャリー入力との間にはキャリー出力スイッチ386が設けられている。キャリー出力スイッチ386は、制御信号CCにより制御されて開閉する。

【0071】図4を参照して、メモリ番込バス群282の4本のデータバス400、402、404、406は、それぞれデータメモリ270、272、274、276に接続される。一方メモリ読出バス群280は8本のメモリ読出バス290～297を含む。データメモリ270は、メモリ読出バス300および301によりメモリ読出バス290、291に接続される。データメモリ272は、メモリ読出バス302、303によりメモ

22

リ読出バス292、293に接続される。データメモリ274はメモリ読出バス304、305によりメモリ読出バス294、295に接続される。データメモリ276はメモリ読出バス306、307によりメモリ読出バス296、297に接続される。メモリ読出バス290～297はそれぞれ分岐して、メモリ読出バス群310、312、314、316として図2に示されるプロセッシングユニット200、202、204、206に接続されている。

【0072】図3に示される構成を有するプロセッシングユニットにより、以下の演算が可能となる。

【0073】(1) データメモリから読出されたデータ同士の間での $n \times n$ ビット乗算、 $n+n$ ビット加算。

【0074】(2) データメモリから読出されたデータと、データバス224Mまたは216M上から与えられるデータとの間の $n \times n$ ビット乗算、 $n+n$ ビット加算。

【0075】(3) データバス224Mと216M上のデータの間での $n \times n$ ビット乗算、 $n+n$ ビット加算。

【0076】(4) 乗算器380の出力する乗算結果の上位nビット(MPM)と、データバス224M上のデータとの間の加算、および乗算器MPYの出力の下位nビット(MPL)とデータバス224L上のデータとの間の加算。

【0077】(5) 乗算結果の上位nビット(MPM)とデータメモリからのデータとの間の加算、乗算結果の下位nビット(MPL)とデータメモリからのデータとの間の加算。

【0078】(6) 乗算結果の上位nビット(MPM)と加算器382の加算結果との間の加算、乗算結果の下位nビット(MPL)と加算器384の加算結果との間の加算(積和演算)。

【0079】(7) データバス224Mと224Lとにより表現される2nビット数とデータバス216Mと216Lとにより表現される2nビット数との間の加算。

【0080】図1～図4に示される第1の実施例の並列処理プロセッサにつき、制御方式を以下に説明する。この第1の実施例の並列処理プロセッサでは、各プロセッシングユニットが独立に制御される。各プロセッシングユニットに対応して命令メモリ260、262、264、266(図2参照)が備えられている。

【0081】各プロセッシングユニットの命令は、データメモリを制御するデータメモリ制御系命令と、プロセッシングユニット内の演算器を制御する演算器制御系命令の少なくとも2系統に分類される。データメモリ制御系命令は、図2に示されるコントロール回路250、252、254、256から対応のデータメモリ290、292、294、296にそれぞれ接続されたバスに出

力されるものである。演算器制御系命令は、各コントロール回路250、252、254、256から、対応のプロセッシングユニット200、202、204、206に向かうバスに出力される。

【0082】図5(a)は、データメモリ制御系命令430の形式を示す。データメモリ制御系命令430は、OPフィールド432と、src0、src1フィールド434、436と、dstフィールド438を含む。

【0083】OPフィールド432は、アドレスモードの指定を行なうためのものである。src0、src1フィールドは、対応のデータメモリから読出すデータの2つのアドレスを指定するためのものである。dstフィールド438は、データメモリに書込むデータのアドレスおよび演算器出力とメモリ書込バス400、402、404、406との間の接続を指定するためのものである。

【0084】図5(b)に、演算器制御系命令450の形式を示す。演算器制御系命令450は、乗算器を制御するためのMPYフィールドと、加算器ADD1、ADD0を制御するためのADD1フィールド、ADD0フィールドと、8本のメモリ読出バス290~297から4個のデータを選択するためのデータを格納するSELフィールドを含む。

【0085】MPYフィールドは、OP0フィールド452と、src00フィールド454と、src01フィールド456と、dst0フィールド458を含む。ADD1フィールドは、OP1フィールド460と、src10フィールド462と、src11フィールド464と、dst1フィールド466を含む。ADD0フィールドは、OP2フィールド468と、src20フィールド470と、src21フィールド472と、dst2フィールド474を含む。

【0086】OP0フィールド452と、OP1フィールド460と、OP2フィールド468とは、それぞれ対応の各演算器の演算内容を指定するためのものである。src00フィールド454と、src01フィールド456と、src10フィールド462と、src11フィールド464と、src20フィールド470と、src21フィールド472とは、各演算器の入力に設けられたセクタを制御するためのデータを格納する。dst0フィールド458と、dst1フィールド466と、dst2フィールド474とは、各演算器とデータバスを接続するためのクロスバースイッチCBS392(図3参照)を制御するためのデータを格納する。

【0087】SELフィールドは、SEL0フィールド476と、SEL1フィールド478と、SEL2フィールド480と、SEL3フィールド482を含む。各フィールドは、それぞれ8本のメモリ読出バスから1

個を選択するためのデータを格納する。したがってSELフィールドにより4つのデータが選択される。

【0088】図5(b)に示されるフィールドのうち、dst0フィールド458は、図示していないがさらに2個のフィールドdst00フィールドとdst01フィールドとに分割され、それぞれ乗算器MPYの2つの出力MPMおよびMPLの出力先を指定するためのデータを格納する。

【0089】図5(b)に示されるフィールド454、456、458(上述の2つのフィールドdst00、dst01)と、フィールド462、464、466、470、472、474とに格納されるデータの値と、各値に対応する各セクタの選択動作とを、以下の第1表~第10表に示す。

【0090】

【表1】

【第1表】

src00	入力信号
0	メモリ
1	データバス224M

【第2表】

src01	入力信号
0	メモリ
1	データバス216M

【0091】

【表2】

(14)

特開平7-64789

25

【第3表】

dst00	出力先
00	データベース210M
01	データベース210L
10	データベース220M
11	データベース220L

10

【第4表】

dst01	出力先
00	データベース210M
01	データベース210L
10	データベース220M
11	データベース220L

20

【0092】

【表3】

26

【第5表】

src00	入力信号
00	AD1
01	メモリ
10	データベース224M
11	-

【第6表】

src11	入力信号
00	メモリ
01	MPM
10	データベース216M
11	-

【第7表】

dst1	出力先
00	データベース210M
01	データベース210L
10	データベース220M
11	データベース220L

30

【0093】

【表4】

[第8表]

src20	人 力 信 号
00	AD0
01	メモリ
10	データベース224L
11	データベース224M

[第9表]

src21	人 力 信 号
00	メモリ
01	MPM
10	データベース216M
11	-

[第10表]

dst2	出 力 先
00	データベース210M
01	データベース210L
10	データベース220M
11	データベース220L

図5(a)に示されるdstフィールド438は、さらに3つのフィールドPU選択フィールド440と、SELW制御フィールド442と、書込アドレス指定フィールド444とに分割される。これらフィールドのうちフィールド440、442はいずれも2ビット長である。これらのフィールド440、442に格納されるデータの値と、各値により選択されるプロセッシングユニットと、SELWにより選択される各演算器の出力との一覧を次の第11表および第12表にそれぞれ示す。

[0094]

[表5]

[第11表]

PU選択フィールド	選択PU
00	PU00
01	PU01
10	PU10
11	PU11

[第12表]

SELW制御フィールド	選択出力
00	MPM
01	MPL
10	AD1
11	AD0

図2に示されるコントロール回路250、252、254、256は、図5、図6に示される命令に従って、第1表～第12表に示されるように各セレクトなどを制御する。

[0095] 図6に、図2に示される命令メモリ260、262、264、266への命令の格納方式を示す。図6において「MCNT」で示されるのはデータメモリ制御系命令であり、「PCNT」で示されるのは演算器制御系命令である。命令メモリ260は、基本的には命令490および492に示されるように、上述のデータメモリ制御系命令と演算器制御系命令とを組にして格納する。図6において命令メモリ216の左側に示す数字(100、101、102、103)は命令メモリ260のアドレスを示す。図6に示される例では100番地にMCNT命令490が、101番地にPCNT命令492がそれぞれ格納されている。プログラムカウンタアドレスが「100」を指している場合には、次の命令として100番地のCNT命令490が読出されることを示す。

[0096] 前述のように基本的には命令はデータメモリ制御系命令MCNTと演算器制御系命令PCNTとを組として取扱っている。しかし、大量のデータに同一の演算を繰り返す場合、各演算器への入力条件および演算内容は最初に一度だけ設定すればよく、データの位置を示すデータメモリのアドレスのみを順次変更していくことで処理できる。そのような場合には、図6のアドレス

29

102、103以下で示されるように、MCNT命令494、496を連続して命令メモリ260に格納しておく。各演算器は次の演算器制御系命令PCNTを受取るまでは、前回に設定された演算器制御系命令に基づいて同じ演算内容を繰り返し実行する。

【0097】以下、この第1の実施例の並列処理プロセッサの動作につき、具体例を用いて順次説明する。以下の例において、各プロセッシングユニット内のセクタ、クロスパススイッチは、第1表～第12表に従い、それぞれの図に示されるような接続を与えるように設定された命令で、予め所望の接続を与えるように切換えられているものとする。

【0098】図7は、各プロセッシングユニット200、202、204、206が、それぞれ独立にnビット精度の演算を行なう例を示す。この場合には、プロセッシングユニット間を接続するデータバスは使用しない。各プロセッシングユニットPU00～PU10(240、242、244、246)において、「×」は乗算器を、「+」は加算器をそれぞれ示す。

【0099】プロセッシングユニット200において 20
は、乗算器への2つの入力とはともにデータメモリからのデータである。乗算器は $n \times n$ ビット構成であり、その出力は2nビットである。この例の場合には、乗算器の出力のうちの上位nビットまたは2nビットに丸め演算を行なった後の上位nビットをデータメモリに出力するように図3に示されるセクタSELW390が設定されるものとする。丸め演算には専用のハードウェアが必要であるが、本願発明とは直接の関連がないため、その図示および説明はここでは行なわない。

【0100】プロセッシングユニット202、204に 30
おいては、それぞれの加算器の一方の2つの入力に、ともにデータメモリからのデータが与えられる。すなわち、各加算器の入力部分のセクタが、データメモリからのデータを選択するように設定されている。プロセッシングユニット206では、積和演算が行なわれている。すなわち、乗算器の2つの入力に、データメモリからの2つのデータが与えられる。乗算器の出力の上位nビットが加算器の一方の入力に与えられ、加算器の出力がその加算器自身の他方の入力に与えられている。

【0101】図7に示される接続例では、各プロセッシ 40
ングユニットからの出力は、メモリ番込バス400、402、404、406(図3参照)を介してデータメモリに番込まれる。これは以下に示す他の接続例でも同様であり、所望の演算結果が得られる乗算器または加算器の出力が、メモリ番込バス400、402、404、406のうちの所望のものに番込まれるように、各プロセッシングユニットのセクタSELW390が制御されるものとする。

【0102】図8は、倍精度2nビットの乗算を行なう 50
場合の、この実施例の並列処理プロセッサのデータバス

30

の接続関係を示す。乗算対象のデータをそれぞれa、bとする。データaの上位nビットと下位nビットとをそれぞれa1、a0と表わす。データbの上位nビットと下位nビットとをそれぞれb1、b0として表わす。すると乗算「 $a \times b$ 」は次のように書ける。

$$\begin{aligned} & \text{【0103】 } (a0 + a1) \times (b0 + b1) \\ & = a0 \times b0 \\ & + a0 \times b1 \\ & + a1 \times b0 \\ & + a1 \times b1 \end{aligned}$$

すなわち、2nビット数同士の乗算 $a \times b$ は、4個のnビット数同士の乗算 $a0 \times b0$ 、 $a0 \times b1$ 、 $a1 \times b0$ 、 $a1 \times b1$ を足し合せたものに分解できる。図8に示される接続例は、2nビットの2つの数a、bを上位nビット、下位nビットに分解して上述の計算を行なうためのものである。

【0104】以下、各プロセッシングユニットごとにその接続関係について説明する。なお、各プロセッシングユニット内の加算器のうち左側がADD1、右側がADD0である。

【0105】プロセッシングユニット200では、乗算器の2つの入力にはデータメモリからの2つのデータが与えられるように各セクタが設定される。乗算器の出力の2nビットのうちの上位nビットが、データバス210Lを介してプロセッシングユニット202の加算器ADD0の一方の入力に接続される。プロセッシングユニット200の加算器ADD1の一方入力には、この加算器ADD1自身の出力が接続される。他方の入力には、データバス216Mを介して、プロセッシングユニット206の加算器ADD1の出力が接続される。プロセッシングユニット200の加算器ADD0の一方の入力には、自分自身の出力が接続される。他方の入力には、データバス216Lを介して、プロセッシングユニット206の加算器ADD0の出力が接続される。加算器ADD0からのキャリーCは加算器ADD1のキャリー入力に与えられる。

【0106】プロセッシングユニット202において 50
は、乗算器の2つの入力に、それぞれデータメモリからの2つのデータが与えられるようにセクタが設定される。乗算器の2nビット出力のうちの上位nビットはプロセッシングユニット202の加算器ADD1の入力の一方に与えられる。加算器ADD1の他方の入力には定数「0」が与えられる。プロセッシングユニット202の乗算器ADD1の出力は、データバス212Mを介してプロセッシングユニット204の加算器ADD1の入力の一方に接続される。プロセッシングユニット202の加算器ADD0の入力の一方には、プロセッシングユニット202の乗算器の下位nビットが与えられる。他方の入力には、前述のとおり、プロセッシングユニット200の乗算器の出力の上位nビットが与えられる。プロ

31

セッシングユニット202においても、加算器ADD0のキャリーCは加算器ADD1に与えられる。

【0107】プロセッシングユニット204においては、乗算器の2つの入力にはデータメモリからの2つのデータが与えられる。乗算器の2nビットの出力のうち上位nビットは加算器ADD1の入力の一方に接続され、下位nビットは加算器ADD0の一方の入力に接続される。加算器ADD1の他方の入力は、データバス212Mを介してプロセッシングユニット202の加算器ADD1の出力に接続される。加算器ADD0の他方の入力は、データバス212Lを介してプロセッシングユニット202の加算器ADD0の出力に接続される。加算器ADD0のキャリー出力Cは加算器ADD1に与えられる。加算器ADD1の出力は、データバス214Lを介してプロセッシングユニット206の加算器ADD0の一方の入力に接続される。

【0108】プロセッシングユニット206においては、乗算器の2つの入力に、データメモリからの2つのデータが与えられるようにセクタが設定される。乗算器の出力のうち上位nビットは加算器ADD1の一方の入力に与えられる。下位nビットは加算器ADD0の一方の入力に接続される。加算器ADD1の他方の入力には定数0が与えられる。加算器ADD0の他方の入力には、前述のようにデータバス214Lを介してプロセッシングユニット204の加算器ADD1の出力が接続される。加算器ADD0の出力はデータバス216Lを介してプロセッシングユニット200の加算器ADD0の一方の入力に接続される。加算器ADD1の出力はデータバス216Mを介してプロセッシングユニット200の加算器ADD1の一方の入力に接続される。加算器ADD0のキャリー出力が、加算器ADD1のキャリー入力に与えられる。

【0109】図8に示される接続において、演算は最低位のビット列の乗算から開始される。プロセッシングユニット200の乗算器には、データメモリから上述の a_0 および b_0 をそれぞれ与える。プロセッシングユニット202の乗算器には、データメモリを介して上述の a_1 および b_1 をそれぞれ与える。プロセッシングユニット204の乗算器には、データメモリから a_0 および b_1 をそれぞれ与える。プロセッシングユニット206の乗算器には、データメモリから a_1 および b_1 をそれぞれ与える。

【0110】プロセッシングユニット200における乗算結果の上位nビットが、データバス210Lを介してプロセッシングユニット202の加算器ADD0に送られる。データバス210Lは、データバス210の下位nビットである。乗算結果の上位nビットをデータバスの下位nビットに出力するということは、実質的にデータをnビット下位にシフトしたことに同等である。

【0111】プロセッシングユニット202の2つの加

32

算器ADD0、ADD1では、乗算器の乗算結果 $a_1 \times b_0$ と、nビット下位にシフトされた $a_0 \times b_0$ との間の2nビットの加算処理が行なわれることになる。加算結果の上位nビットはデータバス212Mを介してプロセッシングユニット204に、下位nビットはデータバス212Lを介してプロセッシングユニット204にそれぞれ与えられる。すなわちこの場合、データのシフトは行なわれない。

【0112】プロセッシングユニット204の乗算器の入力部分のセクタは、データメモリからのデータ a_0 および b_1 を乗算器の入力に与えるように接続が設定される。乗算器の出力のうち上位nビットは加算器ADD1に、下位nビットは加算器ADD0にそれぞれ与えられる。プロセッシングユニット204の2つの加算器の間ではキャリーの入出力が行なわれるため、加算器ADD0と加算器ADD1とは、プロセッシングユニット202の出力する2nビットのデータに対して $a_0 \times b_1$ を加算する2nビット加算処理を行なう。加算処理の上位nビットのみがデータバス214Lを介してプロセッシングユニット206に与えられる。

【0113】プロセッシングユニット206の乗算器の入力部分に設けられたセクタは、データメモリからのデータ a_1 および b_1 を乗算器の2つの入力にそれぞれ与えるように接続が設定される。乗算器の出力の上位nビットは加算器ADD1に与えられる。下位nビットは加算器ADD0に与えられる。プロセッシングユニット204で行なわれた加算結果の上位nビットがデータバス214の下位nビットであるデータバス214Lを介して実質的にnビット下方にシフトされてプロセッシングユニット206の加算器ADD0に与えられる。したがってプロセッシングユニット206では、nビット下位にシフトされたプロセッシングユニット206の出力にさらに $a_1 \times b_1$ を加算する2nビットの加算処理が行なわれる。

【0114】以上のようにして、2nビット同士の数 $a \times b$ の乗算結果が、各クロックごとにプロセッシングユニット206の加算器出力に得られる。

【0115】さらに積和演算をする場合には、図8に示されるようにプロセッシングユニット206の加算器ADD0およびADD1の出力は、それぞれデータバス216Lおよび216Mを介してプロセッシングユニット200の加算器ADD0およびADD1にそれぞれ与えられる。

【0116】この図8に示される接続例では、2nビット精度乗算を行なうのに、すべての乗算器とプロセッシングユニット202、204、206に含まれる加算器とが必要である。このとき同時に、プロセッシングユニット200の加算器による2nビット精度演算も実行できる。したがってこの接続例ではこのプロセッサは2nビット精度の1回の乗算と2nビット精度の1回の加算

とを同時に実行可能である。図9に、この実施例のプロセッサにおいて、FFT（高速フーリエ変換）に用いられるバタフライ演算を行なう場合の接続例を示す。演算はnビット精度とする。バタフライ演算では、3つの複素数a、b、cの間に、 $c + a \times b$ と $c - a \times b$ で表わされる演算を行なう。ar、br、crをそれぞれa、b、cの実数部、ai、bi、ciを同じくa、b、cの虚数部、jを虚数単位とすると、a、b、cはそれぞれ次のように表わされる。

$$[0117] \quad a = ar + j \cdot ai$$

$$b = br + j \cdot bi$$

$$c = cr + j \cdot ci$$

$c + a \times b$ と $c - a \times b$ とは、実数部および虚数部を合せて以下の4個の式により計算できる。

$$[0118]$$

$$cr + (ar \times br - ai \times bi) \quad \dots (1)$$

$$ci + (ar \times bi + ai \times br) \quad \dots (2)$$

$$cr - (ar \times br - ai \times bi) \quad \dots (3)$$

$$ci - (ar \times bi + ai \times br) \quad \dots (4)$$

この式(1)～(4)を求めるためには、見かけ上4回の演算を行なう必要があるが、これらには共通項が存在するので、実際に必要な演算は乗算4回と加算(減算)6回とである。このバタフライ演算を行なう接続は図9に示されるとおりである。

[0119] 図9を参照して、プロセッシングユニット200においては、乗算器の2つの入力にはデータメモリからの2つのデータがそれぞれ与えられる。プロセッシングユニット200の加算器ADD1の一方の入力には乗算器の出力の上位nビットが、他方の入力にはデータバス224Mがそれぞれ接続される。加算器ADD1の出力はデータバス220Mに接続される。

[0120] プロセッシングユニット202の乗算器の2つの入力には、データメモリからの2つのデータが与えられる。プロセッシングユニット202の加算器ADD1の入力の一方には乗算器の出力の上位nビットが、他方の入力にはデータバス226Mがそれぞれ接続される。加算器ADD1の出力はデータバス222Mに接続される。

[0121] プロセッシングユニット204の乗算器の2つの入力には、データメモリからの2つのデータがそれぞれ与えられる。乗算器の出力の上位nビットはデータバス224Mを介してプロセッシングユニット200に接続される。プロセッシングユニット204の加算器ADD1の一方の入力は、データバス220Mを介してプロセッシングユニット200に接続される。加算器ADD1の他方の入力には、データメモリからのデータが与えられる。加算器ADD0の入力も、加算器ADD1の入力と共通に接続される。

[0122] プロセッシングユニット206の乗算器の2つの入力には、データメモリからの2つのデータがそ

れぞれ与えられる。乗算器の出力の上位nビットはデータバス226Mを介してプロセッシングユニット202に接続される。プロセッシングユニット206の2つの加算器ADD0、ADD1のそれぞれの一方の入力はデータバス222Mを介してプロセッシングユニット202に共通に接続される。それぞれの他方の入力には、データメモリからのデータが共通に与えられる。

[0123] プロセッシングユニット204および206の各々において、2個の加算器ADD0およびADD1の一方においては加算処理が、他方においては減算処理がそれぞれ行なわれる。

[0124] 図9に示されるように接続されたプロセッサに、次のようにデータを与えることにより、プロセッシングユニット204および206の加算器の出力として、それぞれ $cr + (ar \cdot br - ai \cdot bi)$ および $cr - (ar \cdot br - ai \cdot bi)$ と、 $ci + (ar \cdot bi + ai \cdot br)$ および $ci - (ar \cdot bi + ai \cdot br)$ が得られる。

[0125] プロセッシングユニット200の乗算器の2つの入力には、データメモリからそれぞれar、brを与える。プロセッシングユニット202の乗算器の2つの入力には、データメモリからそれぞれar、biを与える。プロセッシングユニット204の乗算器の2つの入力には、データメモリからそれぞれai、biを与える。プロセッシングユニット204の2つの加算器ADD0、ADD1の入力の一方には、データメモリからcrを与える。プロセッシングユニット206の乗算器の2つの入力には、データメモリからそれぞれデータai、brを与える。プロセッシングユニット206の加算器ADD0およびADD1の入力の一方には、データメモリからciを与える。

[0126] プロセッシングユニット204の乗算器からプロセッシングユニット200へは、データバス224Mを介して $ai \cdot bi$ が与えられる。プロセッシングユニット200の加算器からプロセッシングユニット204へは、データバス220Mを介して $ar \cdot br - ai \cdot bi$ が与えられる。プロセッシングユニット204の2つの加算器では、crと $ar \cdot br - ai \cdot bi$ の加算および減算がそれぞれ行なわれる。したがって前述のとおり、加算器2つの出力にはそれぞれ、上述の式(1)および(3)が得られる。

[0127] プロセッシングユニット206の乗算器からは、データバス226Mを介して $ai \cdot br$ がプロセッシングユニット202に与えられる。プロセッシングユニット202の加算器ADD1は、 $ar \cdot bi + ai \cdot br$ を出力する。この出力はデータバス222Mを介してプロセッシングユニット206の2つの加算器に与えられる。プロセッシングユニット206の2つの加算器の一方ではciと $ar \cdot bi + ai \cdot br$ との間の加算が、他方では減算が行なわれる。したがってプロセッ

35

シングユニット206の2つの加算器の出力として、上述の式(2)および(4)が得られる。

【0128】なお、この例においても、プロセッシングユニット206、204の乗算器の出力する2nビットデータは、加算器に入力される前に適当な丸め演算によりnビットに丸められるものとする。

【0129】図10に、nビット精度の積和演算を行なう場合の、このプロセッサ内のプロセッシングユニット間のデータバス接続を示す。 $a_i \cdot b_i$ をiを変化させながら加算する演算は、このプロセッサがプロセッシングユニットを4個含むために、4項単位で行なうことができる。まず、図10に示されるプロセッサの接続例を説明する。

【0130】プロセッシングユニット200の乗算器の2つの入力には、データメモリからの2つのデータが与えられる。プロセッシングユニット200の乗算器の出力の上位nビットは加算器ADD1の一方に入力に与えられる。加算器ADD1のうちの他方の入力にはデータバス224Mに接続される。加算器ADD1の出力は、データバス220Mを介してプロセッシングユニット204に接続される。

【0131】プロセッシングユニット202の乗算器の2つの入力には、データメモリからの2つのデータがそれぞれ与えられる。乗算器の出力は加算器ADD1の入力の一方に接続される。加算器ADD1の他方の入力には、データバス226Mが接続される。加算器ADD1の出力は、データバス212Mを介してプロセッシングユニット204に接続される。

【0132】プロセッシングユニット204の乗算器の2つの入力には、データメモリからの2つのデータが与えられる。乗算器の出力の上位nビットは、データバス224Mを介してプロセッシングユニット200の加算器ADD1の一方の入力に接続される。プロセッシングユニット204の加算器ADD1の一方の入力はデータバス212Mに、他方の入力はデータバス220Mにそれぞれ接続される。加算器ADD1の出力はデータバス214Mを介してプロセッシングユニット206に接続される。

【0133】プロセッシングユニット206の乗算器の2つの入力には、データメモリからの2つのデータが入力される。乗算器の出力の上位nビットはデータバス226Mを介してプロセッシングユニット202の加算器ADD1の入力の一方に接続される。プロセッシングユニット206の加算器ADD1の一方の入力はデータバス214Mを介してプロセッシングユニット204に接続される。他方の入力は、加算器ADD1自身の出力に接続される。

【0134】図10に示されるように接続されたプロセッサでは、次のようにしてnビット精度の積和演算が行なわれる。

36

【0135】プロセッシングユニット200、202、204、206に、それぞれデータメモリから (a_0, b_0) 、 (a_1, b_1) 、 (a_2, b_2) 、 (a_3, b_3) を与える。プロセッシングユニット200、202、204、206の乗算器の出力として $a_0 b_0$ 、 $a_1 b_1$ 、 $a_2 b_2$ 、 $a_3 b_3$ がそれぞれ得られる。

【0136】プロセッシングユニット200の出力として $a_0 b_0 + a_2 b_2$ が、プロセッシングユニット202の出力として $a_1 b_1 + a_3 b_3$ がそれぞれ得られる。これらはプロセッシングユニット204の加算器ADD1で加算され、 $a_0 b_0 + a_1 b_1 + a_2 b_2 + a_3 b_3$ が得られる。この例においても、各乗算器出力の2nビットは、加算器に入力される前に適当な丸め演算によりnビットに丸められるものとする。

【0137】図11は、この実施例のプロセッサにおいてnビット精度の積和演算を別の方法により行なう場合の接続例を示す。図10に示されるデータバス接続では、4項ごとの積和を得ていた。これに対し図11に示される接続では、 a_{i+1} 、 b_{i+1} は、 a_i 、 b_i よりも1クロックずつ遅れて入力されるようにされている。その結果、各プロセッシングユニットの出力としては、1項ずつ加算した結果が得られる。最終結果は4項の積和ごとにプロセッシングユニット200の加算器出力に得られるようになっている。

【0138】図11に示される接続は次のようになっている。プロセッシングユニット200においては、乗算器の2つの入力には、データメモリからの2つのデータ (a_0, b_0) が与えられる。乗算器出力の上位nビットはプロセッシングユニット202の加算器ADD1の一方の入力に接続されている。プロセッシングユニット200の加算器ADD1の一方の入力は、データバス216Mに接続されている。他方の入力は、加算器ADD1自身の出力に接続されている。

【0139】プロセッシングユニット202において、乗算器の2つの入力には、データメモリからの2つのデータ (a_1, b_1) が与えられる。乗算器出力の上位nビットはプロセッシングユニット202の加算器ADD1の残りの入力に接続されている。この加算器ADD1の他方の入力は、前述のようにプロセッシングユニット200の乗算器の出力の上位nビットに接続されている。加算器ADD1の出力は、データバス212Mに接続されている。

【0140】プロセッシングユニット204において、乗算器の2つの入力には、データメモリからの2つのデータ (a_2, b_2) が与えられる。乗算器の出力の上位nビットは、プロセッシングユニット204の加算器ADD1の一方の入力に接続される。加算器ADD1の他方の入力は、データバス212Mを介してプロセッシングユニット202の加算器ADD1の出力に接続されている。プロセッシングユニット204の加算器ADD1

37

の出力は、データバス214Mに接続されている。

【0141】プロセッシングユニット206において、乗算器の2つの入力には、データメモリからの2つのデータ(a_3 、 b_3)が与えられる。乗算器出力の上位nビットはプロセッシングユニット206の加算器ADD1の一方の入力に接続されている。加算器ADD1の他方の入力、データバス214Mを介してプロセッシングユニット204に接続されている。プロセッシングユニット206の加算器ADD1の出力は、データバス216Mを介してプロセッシングユニット200の加算器ADD1の一方の入力に接続されている。

【0142】図11に示されるように接続されたプロセッサでは、次のようにして積和演算が行なわれる。まず、プロセッシングユニット200の乗算器の出力として $a_0 \cdot b_0$ が得られる。次にプロセッシングユニット202の加算器の出力として、 $a_0 \cdot b_0 + a_1 \cdot b_1$ が得られる。次にプロセッシングユニット204の加算器の出力として、 $a_0 \cdot b_0 + a_1 \cdot b_1 + a_2 \cdot b_2$ が得られる。またプロセッシングユニット206の出力として、 $a_0 \cdot b_0 + a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3$ が得られる。

【0143】この例においても、各プロセッシングユニット内の乗算器出力の2nビットは、加算器に入力される前に適当な丸め演算によりnビットに丸められるものとする。

【0144】以上のようにこの第1の実施例に係るプロセッサでは、各プロセッシングユニット内のセクタを適切に切替えることにより、幅広い種類の演算を行なうことができる。各プロセッシングユニットの構造は全く同一であるため、プロセッサ内のレイアウトや、接続関係が簡明である。また各プロセッシングユニットの構造が同一であるために、これらプロセッシングユニットを制御するための制御命令に互換性があり、プロセッサの制御が容易になる。また各プロセッシングユニット間での2nビット幅のデータバスを用い、乗算結果の上位nビットを次の演算の下位nビットのデータとして他のプロセッシングユニットに与えることができる。データシフトのための手段を用いずに実質的にデータをnビットシフトすることができる。シフト処理が不要なため処理も高速化される。

【0145】【第2の実施例】図12に示されるのは、本発明の第2の実施例のプロセッサに用いられるプロセッシングユニットの1つ(PU00)である。このプロセッシングユニット520が図3に示される第1の実施例のプロセッシングユニット200と異なるのは、セクタ350からアドレスを受取り、セクタ360の入力に該アドレスのデータを出力するためのROM(読出専用メモリ)530を新たに含むことである。その他の点では、このプロセッシングユニット520は図3に示されるプロセッシングユニット200と全く同一の構成

38

である。したがってその他の部分についての詳しい説明はここでは繰り返さない。

【0146】プロセッサがこのような4個のプロセッシングユニットを含むものと仮定すると、他の3つのプロセッシングユニット(PU01、PU10、PU11)も、このプロセッシングユニット520と全く同一の構成である。

【0147】図12に示される第2の実施例のプロセッサのプロセッシングユニット520では、図3に示される第1の実施例のプロセッシングユニット200の動作に加え、次のような演算処理を行なうことが可能となる。データメモリからROM530のアドレスをこのプロセッシングユニット520に入力するものとする。セクタ350によりそのアドレス信号を選択してROM530に与える。ROM530は、指定されたアドレスに格納されたデータをセクタ360に与える。セクタ360がこのデータをセレクトし乗算器380の一方の入力に与える。

【0148】ROM530に格納するデータとしては、たとえばニュートン・ラプソン(Newton-Raphson)法による除算あるいは開平演算に用いられるデータが考えられる。たとえばニュートン・ラプソン法による除算においては、まず除数の逆数を乗算および加算による漸化式より求め、最後にその逆数に被除数を掛けることにより解を求める。この場合周知のように、漸化式により逆数を求める際の最初の近似値が十分近い値でなければ、漸化式の収束性は悪くなる。この近似値を予めROMに格納しておき、最初の近似値として演算に用いることで、漸化式の収束性が大きく向上し、上述した演算が効率よく行なえる。

【0149】【第3の実施例】本発明に係るプロセッサの第3の実施例の要部を図13に示す。この第3の実施例のプロセッサは、第1の実施例とは異なる方法により命令を実行する。そのためにこの第3の実施例では、命令メモリとして第1の実施例に示される命令メモリ260など(図2参照)に代え、図13および図14に示される命令メモリ542を用いる。図13および図14においては、プロセッシングユニット200を制御するための命令メモリ542のみを示したが、他のプロセッシングユニットを制御するための命令メモリもこの命令メモリ542と全く同一の構成である。

【0150】図13を参照して、このプロセッサのプロセッシングユニット200は、コントロール回路540により直接制御される。コントロール回路540は、命令メモリ542から与えられる演算器制御系命令に従いプロセッシングユニット200を制御する。命令メモリ542はまた、データメモリ270にも接続されており、データメモリ制御系命令をコントロール回路540を介さずに直接データメモリ270に与えるためのものである。

【0151】図14を参照して、命令メモリ542は、複数の命令を格納するためのメモリ550と、メモリ550から読出される2つの命令を格納するための命令レジスタ552と、命令レジスタ552に接続された2つの入力を持つ排他的OR (EXOR) 回路556と、EXOR回路556の出力により制御され、命令レジスタ552に格納された2つの命令のうちの2番めのものをコントロール回路540 (図13参照) に出力するか否かを選択するためのスイッチ554と、EXOR回路556の出力に接続され、メモリ550から次の読出すべき命令のアドレスを所定の論理に従って算出するためのアドレス演算論理558と、アドレス演算論理558の演算結果に従って、メモリ550内の連続する2つの読出アドレスを指定するためのプログラムカウンタ (PC) 560を含む。

【0152】図14においてメモリ550の左側に示される数字 (100、101、102、103) は、各命令が格納されているアドレスを示す。メモリ550に格納された各命令572の先頭 (第1ビット) 570は、その命令がデータメモリ制御系命令であるか、演算器制御系命令であるかを示すフラグとなっている。図14に示される例の場合には第1ビット570が「0」であればメモリ制御系命令であり、「1」であれば演算器制御系命令であるものとする。

【0153】同様に命令レジスタ552も、命令582を格納する領域と、第1ビット580を格納する領域とを有している。命令レジスタ552は、こうした命令を格納する領域を2ヵ所有し、それぞれの第1ビットがEXOR回路556の2つの入力に接続されている。

【0154】図14に示される第3の実施例の命令メモリ542は、第1の実施例に示されるプロセッサの制御をより改善させるためのものである。第1の実施例のプロセッサでは、命令は命令メモリから1個ずつ取出され、それがデータメモリ制御命令か演算器制御系命令かが識別された後、その識別結果にしたがってデータメモリまたは各プロセッシングユニットの演算器の制御が行われていた。しかし、データメモリおよび演算器は相互に独立に動作 (制御) することができる。したがって、データメモリ制御系命令と演算器制御系命令とが続いて命令メモリに格納されている場合には、この2つの命令を同時に実行した方が、順に実行するよりも効率がよい。この第3の実施例はこの点においてプロセッサの制御方法を改良したものである。

【0155】図14を参照して、プログラムカウンタ560のポインタ1 (PC+A) が100番地を、ポインタ2 (PC+A+1) が101番地をそれぞれ指定しているものとする。100番地と101番地の命令は命令レジスタ (IReg) 552に同時に読込まれる。命令レジスタ552に格納された2つの命令の第1ビット580は、EXOR回路556の2つの入力にそれぞれ与

えられる。EXOR回路556の出力は、命令レジスタ552に格納された命令の2つの第1ビット580がともに「1」またはともに「0」であるときには「0」となり、そうでない場合には「1」となる。命令の先頭ビットを、データメモリ制御系命令では「0」に、演算器制御系命令では「1」にしておけば、このEXOR回路の出力により、2つの命令が同一系統かどうかを判断できる。

【0156】EXOR回路の出力が「1」であればスイッチ554は閉じられる。この場合には第2の命令 (図14に示される例の場合にはPCNT命令) がコントロール回路540 (図13参照) に送られる。また、EXOR回路556の出力が0のときにはスイッチ554は開き、第2の命令はコントロール回路540に送られない。一方、第1の命令は常にデータメモリに送られる。

【0157】EXOR回路556の出力が「1」の場合、アドレス演算論理558では、プログラムカウンタ560のAに2を、「0」であればAに1をそれぞれ代入し、プログラムカウンタ560に与える。すなわち、異なる系統の命令が読出された場合には、プログラムカウンタ560のポインタ1 (PC+A) は、メモリ550の102番地を次に指定する。ポインタ2 (PC+A+1) は103番地を指定する。したがって101番地の命令が改めて読出されることはない。

【0158】同一系統の命令が読出された場合には、ポインタ1 (PC+A) は101番地を、ポインタ2 (PC+A+1) は102番地を指定する。この場合には101番地および102番地の命令が同時に読出されて命令レジスタ552に格納され、上述した判断と判断結果に伴う命令の転送とアドレス演算とが行なわれる。

【0159】この第3の実施例では、命令は常に2個ずつ読出され、同一系統の命令であれば2番目の命令は実行されず、単にプログラムカウンタを1増加させてその2番目の命令を含む2つの命令を次に読出す。異なる系統の命令を読出した場合には一度に2つの命令を実行して、プログラムカウンタを2増加させ、読出された2番目の命令の次の2つの命令を次に読出すことになる。データメモリ制御系命令と演算器制御系命令とが連続して格納されている場合、これらを同時に実行することができ、プロセッサの動作効率が向上する。また前述のように各演算器は、直前に入力された演算器制御系命令に従って動作するので、データメモリの制御命令を繰り返し与えることにより、異なるデータに対する同一の演算を効率よく実行することができる。

【0160】【第4の実施例】この発明の第4の実施例に係るプロセッサを図15に示す。図15に示すプロセッサは、8個のプロセッシングユニット600、602、604、606、608、610、612、614と、8つのデータメモリ620、622、624、626、628、630、632、634を含む。各プロ

41

セッシングユニットに付けられた数値は、2進数で表わされたそのプロセッシングユニットのアドレスを示す。データメモリ（DM）に付けられた数値も同様にそのアドレスを示す。

【0161】このプロセッサはさらに、データバス群690、692、694、696、698、700、702、704、706により各データメモリ620、622、624、626、628、630、632、634に接続されたメモリ書込バス群680と、データメモリ620、622、624、626に接続され、かつデータバス群710、712、714、716によりそれぞれプロセッシングユニット600、602、604、606に接続されたメモリ読出バス群682と、データメモリ628、630、632、634に接続され、かつデータバス群718、720、722、724によりそれぞれプロセッシングユニット608、610、612、614に接続されたメモリ読出バス群684とを含む。

【0162】さらにこのプロセッサでは、第1の実施例におけると同様に、アドレスが1ずつ異なるプロセッシングユニットを順次円環状に単方向に接続するためのデータバス640、642、644、646、648、650、652、654と、アドレスが2だけ異なるプロセッシングユニットを双方向に接続するためのデータバス群660、662、664、666、668、670、672、674とを含む。データバス群660、662、664、666、668、670、672、674は、それぞれ逆方向を向いた1対のデータバスを含んでいる。

【0163】図15に示される構成では、各プロセッシングユニットにおいて、他のプロセッシングユニットとの間のデータバスとしては、3組の入力と3組の出力とがある。これは各プロセッシングユニット共通である。またプロセッシングユニットとデータメモリとはそれぞれ4個ずつの2つのグループに分類されている。第1のグループはプロセッシングユニット600、602、604、606とデータメモリ620、622、624、626とを含み、第2のグループはプロセッシングユニット608、610、612、614と、データメモリ628、630、632、634とを含む。各グループのデータメモリとプロセッシングユニットとはそれぞれメモリ読出バス群682、684により接続されている。また各プロセッシングユニットとメモリ書込バス680とは、図示されていないメモリ書込バス群により接続され、各プロセッシングユニットから任意のデータメモリに対してデータを書込むことができる。このようにすることにより、グループ間の通信をデータメモリを介して行なうことができる。

【0164】図15に示されるようにデータメモリをグループ化するの、データメモリの読出バスをできるだ

42

け短くするためである。データメモリの読出バスは長くなるとスピードが遅くなる。したがって、この実施例のようにプロセッシングユニット4個単位でグループ化してその長さを短くした方が動作速度上有利な場合が多い。また実際の応用における演算では、倍精度演算、バタフライ演算など、プロセッシングユニットを4項単位で使用する演算要求が多い。したがって図15に示されるようにプロセッシングユニットとデータメモリとを4項単位でグループ化することにより、実際の、かつ幅広い種類の演算に対応できるプロセッサを得ることができる。

【0165】図15に示されるプロセッサでは、それぞれのグループにおいて倍精度の1乗算と1加算、バタフライ演算、4項ごとの積和演算がそれぞれ可能である。各プロセッシングユニットで用いる入力データはそのグループ内にあるので、グループ内でデータを共有できる。したがって図15に示される構成と異なり、データメモリからのデータ読出バスをそのグループ内で閉じるようにしてもよい。

【0166】図15に示されるプロセッサでは、各プロセッシングユニットの構造は相互に全く同一である。したがって、各プロセッシングユニットを制御する制御命令には互換性がある。またプロセッサを作製する上で、プロセッシングユニットやデータメモリのレイアウトが単純でよいという利点がある。また、第1の実施例と同様に、各プロセッシングユニット内のセレクトを、所望の演算を実現できるように切換えることにより、複数のプロセッシングユニットを用いた複雑な演算を、従来のものよりもより多種類実行することができる。

【0167】もちろん、いずれのプロセッシングユニットも任意のデータメモリからデータを読出せるようにデータメモリ読出バス群とプロセッシングユニットとを接続することも考えられる。たとえば8項ごとの積和演算は、図15に示されるようにグループ化せず8つのプロセッシングユニットを1つのグループとして構成した方が効率はよい。しかしその場合には、データメモリからの読出速度が低下するおそれがある。また、4項ごとの演算を行なうような場合には、図15に示されるようにグループ化した方が好ましい。

【0168】

【発明の効果】以上のように請求項1に記載の並列処理のためのプロセッシングユニットは制御手段の制御により種々の演算が可能のため、汎用的でありかつ制御手段の制御は比較的単純でよい。プロセッシングユニットに接続される入出力データバスの数がすべてのプロセッシングユニットで同一であり、また各プロセッシングユニットの構造が同一であるために、これらを複数個組合わせて並列処理プロセッサを作製するときのプロセッシングユニット相互の接続が容易であり、それらの制御に互換性を持たせることができる。

【0169】その結果、より幅広い種類の演算を簡単に制御で実現できる、並列処理のためのプロセッシングユニットを提供できる。

【0170】請求項2に記載のプロセッシングユニットは、 $n \times n$ ビットの乗算器と $n + n$ ビットの2つの加算器を用いて、大量のデータ処理が要求される演算における典型的な処理を行なうことができる。

【0171】その結果、より幅広い種類の大量の演算を簡単な制御で実現できる、並列処理のためのプロセッシングユニットを提供できる。

【0172】請求項3に記載のプロセッシングユニットでは、 $2n$ ビットのデータ同士の演算を、それぞれ上位 n ビットと下位 n ビットとに分けて $n \times n$ ビットの乗算器および $n + n$ ビットの加算器を用いて効率よく行なうことができる。

【0173】その結果、倍精度演算を含むより幅広い種類の演算を簡単な制御で実現できる、並列処理のためのプロセッシングユニットを提供できる。

【0174】請求項4に記載のプロセッシングユニットでは、2つの加算器の一方のキャリー出力を他方のキャリー入力とする場合と、しない場合とを選択できる。したがって、 $2n$ ビットのデータ同士の加算と、 n ビットのデータ同士の2つの別個の加算とを効率よく実行することができる。

【0175】その結果、倍精度の加算処理を含むより幅広い種類の大量の演算を簡単な制御で実現できる、並列処理のためのプロセッシングユニットを提供できる。

【0176】請求項5に記載のプロセッシングユニットの加算器では、自己の加算結果を用いる演算、たとえば積和演算と、データバス経由のデータを用いた加算との双方を効率よく行なうことができる。

【0177】その結果、積和演算を含むより幅広い種類の大量の演算を簡単な制御で実現できる、並列処理のためのプロセッシングユニットを提供できる。

【0178】請求項6に記載のプロセッシングユニットの加算器は、乗算結果を加算の入力に用いる演算、たとえば積和演算と、データバス経由のデータを用いた加算との双方を効率よく行なうことができる。

【0179】その結果、積和演算を含むより幅広い種類の大量の演算を簡単な制御で実現できる、並列処理のためのプロセッシングユニットを提供できる。

【0180】請求項7に記載のプロセッシングユニットの加算器は、乗算結果や加算結果を加算の入力に用いる演算、たとえば積和演算と、データバス経由のデータを用いた加算との双方を効率よく行なうことができる。

【0181】その結果、積和演算を含むより幅広い種類の大量の演算を簡単な制御で実現できる、並列処理のためのプロセッシングユニットを提供できる。

【0182】請求項8に記載のプロセッシングユニットの乗算器では、記憶手段に格納されたデータを用いるこ

とにより処理の高速化を図れる処理、たとえばニュートン・ラプソン法による除算や、開平演算などを効率よく行なうための構成と、通常の演算を行なうための構成とを自由に選択できる。

【0183】その結果、ニュートン・ラプソン法や開平演算を含むより幅広い種類の大量の演算を簡単な制御で効率良く実現できる、並列処理のためのプロセッシングユニットを提供できる。

【0184】請求項9に記載の並列処理プロセッサでは、各プロセッシングユニットは、最低でも2以上の同じ数の入力と出力とを有する。入出力の数が多く、演算の自由度が高くなる。また、各プロセッシングユニットは同一の構造を有するため、プロセッサのレイアウトおよび制御の複雑さが低減される。

【0185】その結果、より幅広い種類の演算を簡単な制御で実現できる、簡単な構成の並列処理のためのプロセッサを提供できる。

【0186】請求項10に記載の並列処理プロセッサの各プロセッシングユニットは、データ記憶手段からのデータと、他のプロセッシングユニットからのデータとのいずれにも、複数個の演算手段を用いた所定の処理を行ない、任意の演算手段の出力を他のプロセッシングユニットと、データ記憶手段との任意のものに出力できる。したがって、複数の同一構造のプロセッシングユニットを用いた多くの種類の演算を、互換性のある制御命令を用いて実行できる。

【0187】その結果、より幅広い種類の演算を簡単な制御で実現できる、並列処理のためのプロセッサを提供できる。

【0188】請求項11に記載の並列処理プロセッサでは、データ記憶手段からプロセッシングユニットへの読出データバスは、プロセッシングユニットをグループ化せず全体に接続する場合と比較して短くすみ、プロセッシングユニットへのデータの機器が高速に行なえる。また1グループのプロセッシングユニットで共通のデータを用いた演算を実行できる。さらに各プロセッシングユニットからは任意のデータ記憶手段にデータを書込めるので、グループ間のデータ交換も可能で、複数のプロセッシングユニットを用いた多彩な演算を実行できる。また各プロセッシングユニットは同一構成で、互換性のある制御命令で制御することができる。

【0189】その結果、より幅広い種類の演算を簡単な制御で高速に実行できる、並列処理のためのプロセッサを提供できる。

【0190】請求項12に記載の並列処理プロセッサの動作方法では、演算結果の上位 n ビットまたは下位 n ビットは、新たな演算のためのデータの低位 n ビットまたは上位 n ビットとして乗算手段および加算手段に与えることができ、多様な演算を行なうことができる。また、2つの加算手段の間でキャリーの入出力を行なうかどう

かを選択できるので、 n ビット精度の2つの加算処理と $2n$ ビット精度の1つの加算処理とを切換えて行なうことができる。したがって、非常に自由度の高い処理を行なうことができる。また各プロセッシングユニットの構成は同一で、かつ互換性のある制御命令で制御できる。

【0191】その結果、より幅広い種類の演算を簡単な制御で実行できる、並列処理のためのプロセッサの動作方法を提供できる。

【0192】請求項13に記載の並列処理プロセッサの動作方法では、あるプロセッシングユニット内の乗算手段または加算手段の出力の上位 n ビットまたは下位 n ビットが、他のプロセッシングユニット内の乗算手段または加算手段の下位 n ビットと上位 n ビットとの任意の一方に☐されるようにデータバスと乗算手段または加算手段との接続を設定できる。シフト手段を用いることなく多様な演算を行なうことが可能である。またシフト手段を用いる場合よりも動作が高速で、制御も単純である。

【0193】その結果、より幅広い種類の演算を簡単な制御で高速に実行できる、並列処理のためのプロセッサの動作方法を提供できる。

【0194】請求項14に記載の並列処理プロセッサの動作方法では、あるプロセッシングユニット内の乗算手段または加算手段の出力の上位 n ビットが、他のプロセッシングユニット内の乗算手段または加算手段の下位 n ビットに☐されるようにデータバスと乗算手段または加算手段との接続を設定できる。シフト手段を用いることなくデータを実質的にシフトでき、シフトのための時間も必要としない。シフト手段を用いることなく多様な演算を行なうことが可能である。またシフト手段を用いる場合よりも動作が高速で制御も単純である。

【0195】その結果、より幅広い種類の演算を簡単な制御で高速に実行できる、並列処理のためのプロセッサの動作方法を提供できる。

【0196】請求項15に記載の並列処理プロセッサでは、複雑な演算を各プロセッシングユニットごとに制御すればよく、並列処理プロセッサ全体の制御が容易になる。また、各プロセッシングユニットでは、制御手段により多様な演算処理を行なえ、かつ複数のプロセッシングユニットを組合わせることにより、プロセッサ全体としてさらに従来より幅広い処理を実現できる。

【0197】その結果、より幅広い種類の演算を簡単な制御で実現できる、並列処理のためのプロセッサを提供できる。

【0198】請求項16に記載の並列処理プロセッサでは、データ記憶手段から出力されるデータを変化させつつ一定の命令を繰り返して行なう場合などに、データ記憶手段を制御する命令のみを、その内容を変化させつつ繰り返し発行すれば、演算手段のための異なる制御命令をいくつも発行する必要がない。したがって、大量のデー

タに対する同一の演算を行なう場合の制御が容易である。また、制御命令を組合わせることで、各プロセッシングユニットにおいて多様な演算処理を行なうことができる。複数のプロセッシングユニットを組合わせることができ、さらにより複雑な演算を実現できる。

【0199】その結果、より幅広い種類の演算を簡単な制御で実現できる、並列処理のためのプロセッサを提供できる。

【0200】請求項17に記載の並列処理プロセッサでは、データ記憶手段の各々の2つの読出アドレスからの2つのデータを読出して所定の演算を行ない、結果を1つの書込アドレスにより指定された位置に書込できる。複数のプロセッシングユニットの間で、データ記憶手段を介してデータを授受しつつ、複雑な演算を実行できる。各プロセッシングユニットは同一構成であり、そのレイアウトは単純でよい。しかも互換性のある制御命令でプロセッシングユニットを制御でき、プロセッサの制御が簡略になる。

【0201】その結果、より幅広い種類の演算を簡単な制御で実現できる、並列処理のためのプロセッサを提供できる。

【0202】請求項18に記載の並列処理プロセッサでは、データ記憶手段から出力されるデータを変えつつ一定の命令を繰り返して行なう場合などに、データ記憶手段を制御する命令のみを、その内容を変化させつつ繰り返し発行すれば、演算手段のための同一の制御命令を繰り返し発行する必要がない。したがって大量のデータを処理する場合、プロセッサ全体の制御が単純になる。

【0203】その結果、より幅広い種類の大量の演算を簡単な制御で実現できる、並列処理のためのプロセッサを提供できる。

【0204】請求項19に記載の並列処理プロセッサでは、命令メモリから読出された2つの命令が同一の系統に属すれば、それらは同時に実行される。異なる系統であれば通常と同様の処理が行なわれる。一度に制御命令を1つしか読出さない場合と比較して、命令の実行速度は向上する。

【0205】その結果、より幅広い種類の演算を簡単な制御で高速に実行できる、並列処理のためのプロセッサを提供できる。

【図面の簡単な説明】

【図1】本発明の第1の実施例に係るプロセッサの構成を示す模式的ブロック図である。

【図2】第1の実施例のプロセッサのブロック図である。

【図3】第1の実施例のプロセッシングユニットのブロック図である。

【図4】第1の実施例におけるデータメモリとメモリ読出データバス群との間の接続を示すブロック図である。

【図5】制御命令の構成を示す模式図である。

47

【図6】命令メモリにおける命令の格納状態を示す模式的ブロック図である。

【図7】本発明の第1の実施例のプロセッサによる第1の接続例を示す模式的ブロック図である。

【図8】本発明の第1の実施例のプロセッサによる第2の接続例を示す模式的ブロック図である。

【図9】本発明の第1の実施例のプロセッサによる第3の接続例を示す模式的ブロック図である。

【図10】本発明の第1の実施例のプロセッサによる第4の接続例を示す模式的ブロック図である。

【図11】本発明の第1の実施例のプロセッサによる第5の接続例を示す模式的ブロック図である。

【図12】本発明の第2の実施例のプロセッサのプロセッシングユニットのブロック図である。

【図13】本発明の第3の実施例のプロセッサの要部のブロック図である。

【図14】本発明の第3の実施例における命令メモリの模式的ブロック図である。

【図15】本発明の第4の実施例のプロセッサの構成を示す模式的ブロック図である。

【図16】従来の並列処理プロセッサを示す模式的ブロック図である。

【図17】図16に示す従来のプロセッサの各プロセッシングユニットの構成を示すブロック図である。

10

20

48

【図18】図16および図17に示される従来のプロセッサのプロセッシングユニットの接続例を示す模式的ブロック図である。

【図19】従来のプロセッサのプロセッシングユニット相互の間の接続例を示す模式的ブロック図である。

【符号の説明】

200、202、204、206 プロセッシングユニット

210、212、214、216 単方向入力データバス

220、222、224、226 単方向入力データバス

250、252、254、256 コントロール回路

260、262、264、266 命令メモリ

280 メモリ読出データバス群

282 メモリ書込データバス群

270、272、274、276 データメモリ

350、352、354、356 セレクタ

360、362、364、366、368、370 セレクタ

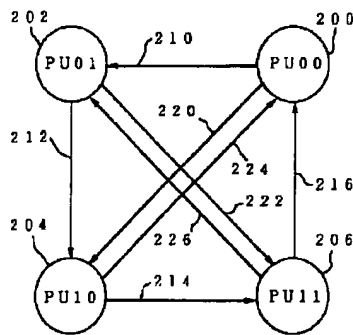
380 乗算器

382、384 加算器

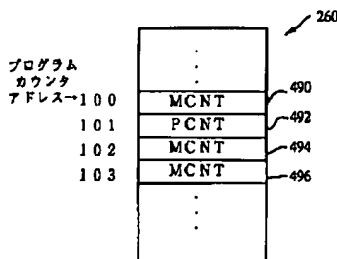
390 セレクタ

392 クロスバースイッチ

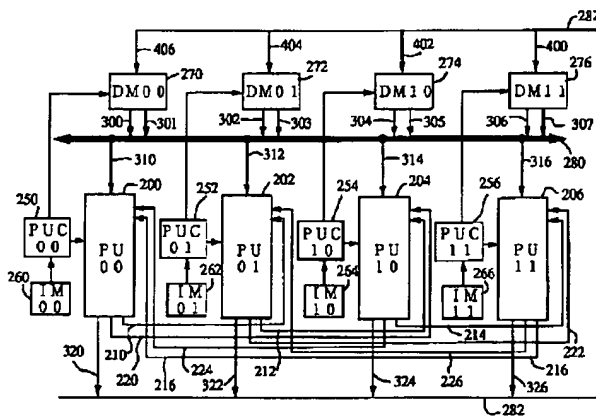
【図1】



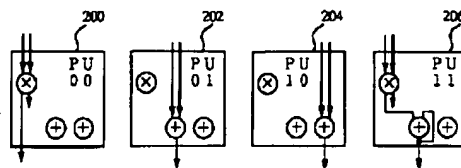
【図6】



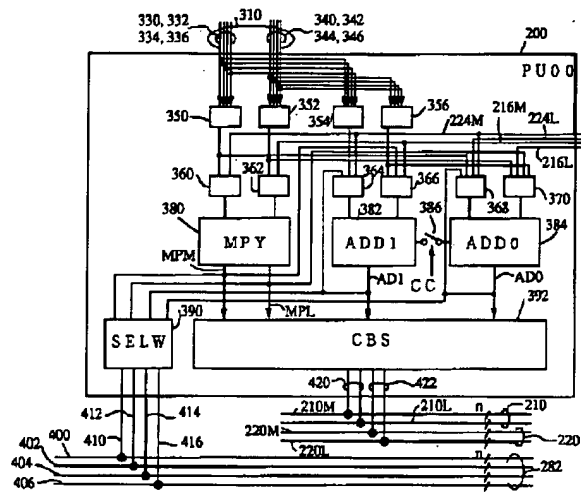
【図2】



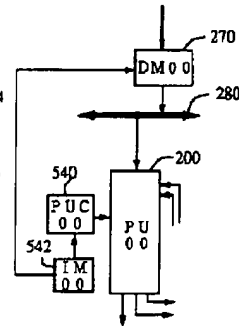
【図7】



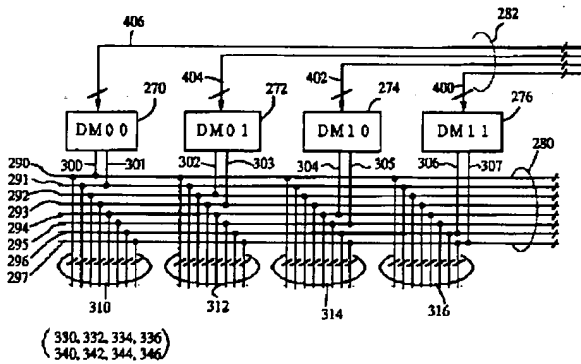
【図3】



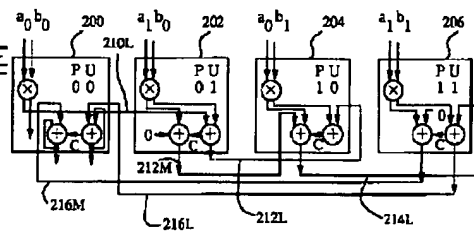
【図13】



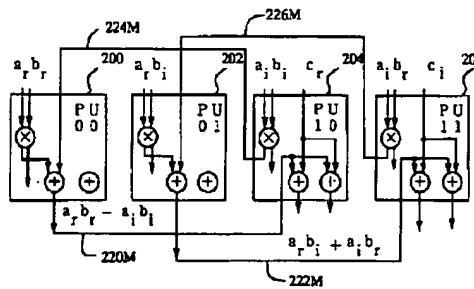
【図4】



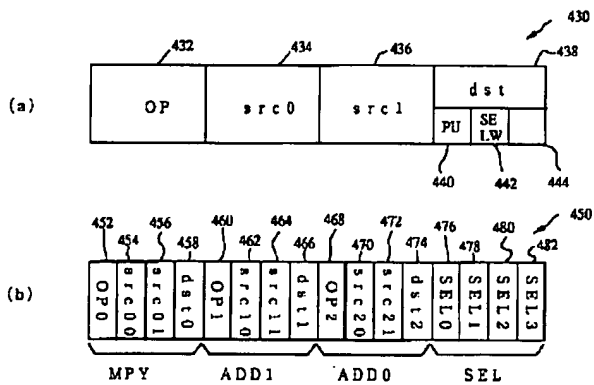
【図8】



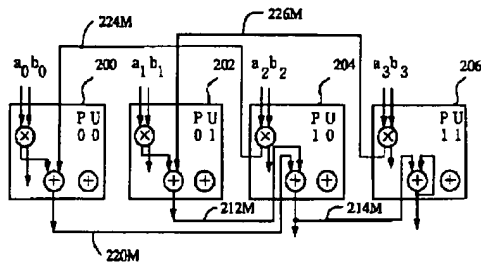
【図9】



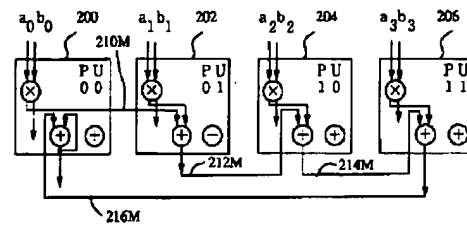
【図5】



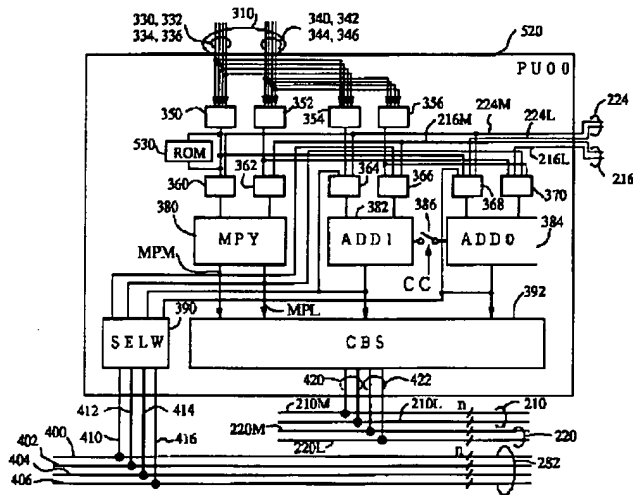
【図10】



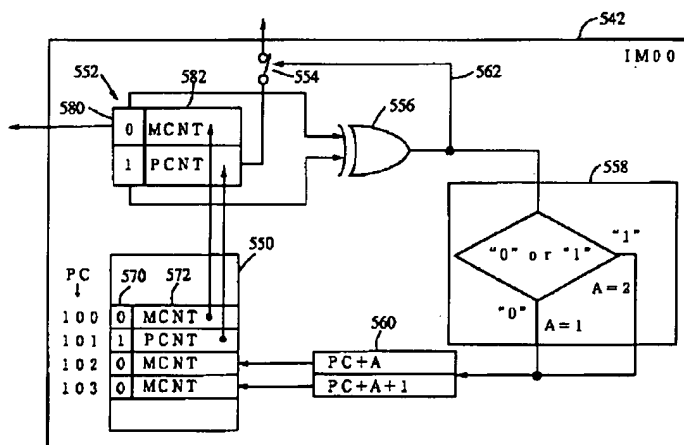
【図11】



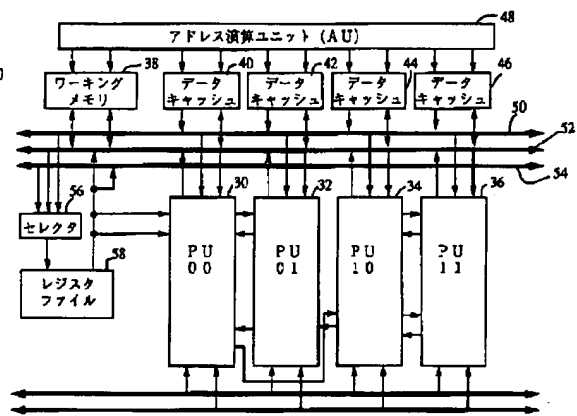
【図12】



【図14】



【图 16】



The diagram illustrates a four-processor system with the following components and connections:

- Processors:** PU00, PU01, PU10, and PU11.
- Internal Components:** Each processor contains an ALU, an MPY (Multiplier), an ADD (Adder), and an LPM (Local Program Memory) block.
- Connections:**
 - PU00 and PU01 are connected to PU10 and PU11 via a central bus system.
 - Each processor has its own LPM block (LPM00, LPM01, LPM10, LPM11).
 - Data paths and control lines are shown between the components and between the processors.

【図19】

